网络空间安全学院 2019 — 2020 学年第 2 学期
《信息系统安全》 考试试卷

A 卷　　开卷　　考试时间：2020 年 6 月 29 日

专业 信息安全 班级 _____ 学号 _____ 姓名

题号	一	二	三	四	五	六	七	总分	核对人
题分	15	30	11	12	10	12	10	100	
得分									

得分	评卷人

一、判断题（判断以下各题的说法是否正确，正确的打 √，错误的的打×，每小题 1.5 分，共 15 分）

（ √ ）1. 某项系统功能以共享库的形式提供给用户，由用户按照各自的需要调用，而不是以系统服务的形式提供给全体用户，这种做法主要是考虑最小特权原则。

（ √ ）2. 自主访问控制 DAC 不能抵御特洛伊木马攻击。

（ √ ）3. BLP 安全模型所指定的原则是利用“不上读”、“不下写”来保证数据的机密性。

（ × ）4. 系统调用介入（System call interposition）是一种在现代操作系统中加快系统调用处理速度的方法。

（ × ）5. 用于防止缓冲区溢出攻击的 stack canaries 可以应用于可执行文件，而无需重新编译代码。

（ × ）6. 攻击者不可能在遵循控制流图（CFG）的前提下，进行控制流劫持攻击。

（ √ ）7. 使用参数化 SQL 语句实现 Web 应用程序可有效防御 SQL 注入攻击。

（ √ ）8. 最小特权原则可以用来有效的抵御基于缓冲区溢出的攻击。

（ √ ）9. 恶意软件可以通过检测虚拟机管理器 VMM 是否存在，以拒绝在虚拟机环境中运行。

（ × ）10.联盟链通常使用 POW 共识机制。

得分	评卷人

二、简答题（简要回答以下问题，每小题 5 分，共 30 分）

1. 在 Linux 操作系统中可以使用 setuid( )系统调用来撤销进程的权限，但在撤销进程权限后，可能存在 capability 泄露问题，请简要说明该问题及其原因。

答：capability 是 Linux 操作系统中一种权限机制，它允许进程在拥有少量权限的情况下执行特定的操作。在 Linux 操作系统中，每个进程都拥有一组 capability，这些 capability 决定了进程可以执行的操作。使用 setuid() 系统调用撤销进程权限后，该进程的 capability 仍然保持不变，这可能会导致 capability 泄露问题。

2. 缓冲区溢出攻击和 XSS 攻击都利用不安全的系统设计，即容易受到“Mixing data and code”的影响。对于这两种攻击，请说明用户数据是什么以及它如何影响程序控制的。

缓冲区溢出攻击中用户数据是栈上的数据，而栈上数据混杂了局部变量数据以及函数返回地址（程序控制流），缓冲区溢出攻击通过篡改函数返回地址达到程序控制。

XSS 攻击中用户数据是网页上的Html数据，其中混杂了正常的富文本以及恶意的js代码，通过让受害者不经意间执行相关恶意js代码影响程序控制。

3. 什么是隐蔽通道？它是如何被用于在单个机器上运行的多个相互隔离的 VM 之间泄漏信息的。

隐秘通道：隔离组件间的非预期通信信道。

由于所以虚拟机共用同一底层硬件，可以通过在某一指定时间是否占用大量CPU计算资源/内存资源来传递01的1比特信息。

4. 以下哪种技术可以帮助防御基于堆的控制劫持攻击。简要说明这些的技术如何提供帮助。

Stackguard, ASLR, DEP, StackShield, CFI

Stackguard：不能防堆的控制劫持，主要用于防栈溢出。
ASLR：可以防，让攻击者难以找到目标堆区域的起始地址。
DEP：可以防，防止攻击者利用堆中的数据来执行恶意代码。
StackShield：可以防，防止攻击者利用堆中的数据来覆盖栈上的数据。
CFI：可以防，防止攻击者利用堆溢出漏洞来修改程序的控制流。

5. 考虑一个实现电话拨号程序的网站 xyz.com。 当用户输入要拨打的电话号码时，浏览器将打开一个新窗口，其中包含以下定义了监听 postMessage 事件的 Javascript：

```
function receiveMessage(event) {
        // event.data is a phone number from sender
        initiatePhoneCallTo(event.data);
}
window.addEventListener("message", receiveMessage, false);
```

然后，父页面将 postMessage 发送到此窗口，从而激活 receiveMessage 函数以发起呼叫。 请说明恶意网站如何导致访问者向任意电话号码发起电话呼叫（假设访问者已登录到他的 xyz.com 帐户）。

利用CSRF，伪造一个含message的post请求。

```
<html>

<body>
    <script type="text/javascript">
        window.onload = function () {
            var payload;
            payload += "<input type='hidden' name='message' value='xxx'>";

            var p = document.createElement("form");
            p.action = "http://xyz.com";
            p.innerHTML = payload;
            p.target = "_self";
            p.method = "post";
            document.body.appendChild(p);
            p.submit();
        }
    </script>
</body>

</html>
```

6. 以下左图间接跳转会受到一种称为”Branch target injection”的攻击（Spectre 的一种攻击形式），即：jmp 指令查询间接分支预测器（Indirect Branch Predictor），间接分支预测器会依据 BTB（Branch Target Buffer）中内容进行预测，如果预先对 BTB 中的内容进行操纵，那么会使得程序跳转到攻击者给定的目标。以下右图为针对该攻击的防御方法 Retpoline，请简要解释该方法是如何解决 Branch target injection 攻击的？

```
jmp *[eax]
```

```
call l5
l2: pause
lfence
jmp l2
l5: add rsp, 8
push [eax];
ret
```

不会,retpoline是一个返回蹦床,它使用永远不会执行的无限循环来阻止CPU推测间接跳转的目标.

得分	评卷人

三、（11 分）下表是一个 Linux 目录中的文件信息。

Permissions	Owner	Group	Size	Last Update	File Name
-rws-----x	dave	gdev	1452306	Nov 03 21:11	gtool
drwxrwxrwt	dave	gdev	1452306	Nov 03 21:11	gdata
-rwx--x--x	alice	alice	214768	Nov 03 09:36	setup
-rw-r-----	alice	pcrack	12486	Dec 04 11:00	sourcg
-rw-r--r--	dave	pcrack	14257	Oct 02 18:44	config
-rw--wxr--	root	pcrack	176704	Nov 01 12:23	hosts

（1）列出alice可以write的文件的名称；（2分）

（2）列出dave可以read的文件的名称；（2分）

（3）假设alice执行程序gtool，列出相应进程可以execute的文件的名称；（2分）

（4）假设dave执行程序setup，列出相应进程可以write的文件的名称；（2分）

（5）alice是否可以删除gdata目录内部的文件？为什么？（3分）（提示：sticky位）

1. gdata setup sourcg
2.gtool gdata config hosts
3.gdata setup
4.gtool gdata config
5.不能，只有文件的创建者和超级用户才能删除这些文件。

得分	评卷人

四、（12 分）

已知以下一段源代码片段：

```
char tmpbuf[512];
char outbuf[512];
snprintf (tmpbuf, sizeof (tmpbuf), "foo: %s", user);
tmpbuf[sizeof (tmpbuf) - 1] = '\0';
sprintf (outbuf, tmpbuffer);
```

其中，snprintf 函数定义如下，该函数通过 size 参数来约束格式化字符串的长度：

```
int snprintf(char *str, size_t size, const char *format, ...);
```

(1) 分析上述源代码片段存在什么问题，并指出该问题的后果；（4 分）

(2) 请具体说明如何提供字符串 user 来利用该问题，并绕过格式化字符串的长度限制;（4 分）

(3) 在关闭地址随机化且栈不可执行保护未开启的情况下，图示说明构造恶意输入需要完成哪些任务，并说明如何增加跳转到 shellcode 正确地址的机会。（4 分）

1.sprintf (outbuf, tmpbuffer); 有格式化字符串漏洞
2.%.{num}X,可以格式化字符串长度到{num}，绕过长度限制
3.计算outbuf到ebp栈帧的偏移，泄露system、“ /bin/sh” 的地址，修改返回地址为system函数地址，修改system函数参数为“ /bin/sh” 。

得分	评卷人

五、（10 分）

Kerberos 涉及三个双向消息交换，一个位于客户端（Client）和密钥分发中心（KDC）之间，一个位于 Client 和票证授予服务（TGS）之间，另一个位于客户端和服务器之间（S） 客户。

（1）简要说明三个消息交换（Client 与 KDC，Client 与 TGS，Client 与 S 之间）的目的；（4 分）

（2）相对于基于 PKI 的认证系统来说，为什么说 Kerberos 系统的可用性（usability）更好？（2 分）

（3）说明 Kerberos 中是如何实现向客户端认证服务器的？（4 分）

1

Client与KDC：Client获取一次TGS票据
Client与TGS：Client使用TGS票据获取每个网络服务的服务票据和短期密钥
Client与S：使用短期密钥以及服务票据登录S

2

Kerberos 的使用是透明的，是因为 Kerberos 实现了单点登录（Single Sign-On）功能，用户只需要在登录时输入一次用户名和密码，就可以在整个网络中访问需要认证的资源，而无需在每个资源上都输入用户名和密码。这种透明的认证方式可以提高用户的使用便捷性和工作效率，同时也可以降低系统管理员的管理成本。

3 略，见PPT

得分	评卷人

六、（12 分）

Seccomp 是在大多数 Linux 发行版中启用的一项内核功能。它对线程所能调用的系统调用进行了限制，限制为：read()，write()，exit()，sigreturn()等 4 个系统调用。如果某线程调用任何其它系统调用，则包括该进程（此线程所在的进程）中的所有线程在内的整个进程都将终止。这种对线程的约束，对于安全性来说是非常理想的。

Chrome 将 Seccomp 用于隔离 HTML 渲染器（HTML Renderers），对 HTML 渲染器的系统调用进行限制，从而使得 HTML 渲染器不会影响 Chrome 控制器。但是，HTML 渲染器需要支持的不仅仅是 Seccomp 允许的四个系统调用。为了解决这个问题，Chrome 在隔离的 HTML 渲染器线程的同一进程地址空间中运行了一个额外的可信线程（Trusted thread）。只要渲染器线程需要进行非授权的系统调用，它就会将请求发送给该可信线程，该线程检查系统调用的参数，如果授权，它则代表渲染器进行调用。通过在同一进程下运行这两个线程，Chrome 在隔离渲染器的前提下实现了良好的整体性能。但是，同一地址空间下运行意味着渲染器线程可以读取和修改可信线程的内存。

假设渲染器线程已经被入侵，并正在运行恶意代码，那么：

（1）假设渲染器线程请求可信线程代表它打开文件，并且文件名存储在内存中。可信线程从内存中读取文件名，验证它是否指向良性文件（例如在/tmp 中），然后发出系统调用以打开文件。 具体来说，进行如下调用：

```
if (benign(filename))
    fp = open(filename, "r");
```

恶意渲染器如何利用此设计打开受保护的文件？ 这种类型的漏洞称为什么？（4 分） TOCTOU漏洞，benign检测完之后将文件link到需要打开的文件

（2）提出一个解决上述问题的方法，并进行解释？ 重复Check、sticky符号链接保护

（3）上述代码的另一个问题是，渲染器线程和可信线程在相同的地址空间，恶意的渲染器线程可以轻松地操纵可信线程的堆栈。请提出一个解决此问题的方法，并进行解释？（4 分） 不会，分成两个进程，使用两套地址空间吧？

得分	评卷人

七、(10 分) 张三 2020 年 4 月 1 号挖到了一个区块（交易号 #1213），包含 12.5 个币，3 天后李四挖到了一个区块（交易号 #1223）并打包了张三转 5 个币给女婿王五的交易（交易号 #1313），一周后张三庆祝女儿张倩生日张三把剩下的币转给了女儿（交易号 #1420），又过了 3 天张倩和李四吃饭 AA 付饭钱共计 12 个币给了赵六（交易号 #1460）。

根据 UTXO 机制画出整个支付流程。

不考区块链