CSE 40567 / 60567: Computer Security

Security Basics 4 / Cryptography 1
Homework #1 Due Tonight at 11:59PM Eastern Time

See Assignments Page on the course website for details
Man-in-the-Middle Attack

- General category of attacks
- Active attacks are a lot more powerful than passive ones
- A “man-in-the-middle” can modify, delete, and create new messages
Local Exploitation of Bugs

• Target vulnerabilities on a single host system

• Requires some level of access to that system

• Goal is usually privilege escalation
  ‣ Could also target data and meta-data
Privilege Escalation

Some accounts have more privilege than others.

Example: UID 0 in Unix is the super user.

This attack exploits a bug, design flaw or configuration problem in an OS or application.

Linux Kernel 2.6.34 - CAP_SYS_ADMIN x86 - Local Privilege Escalation Exploit

Image Credit: https://www.exploit-db.com/exploits/15916/
Remote Exploitation of Bugs

- Target vulnerabilities on a server
- Does not require some previous level of access
- Goal is unauthorized access
  - Could also target data and meta-data

Example:
ProFTPD IAC - Remote Root Exploit
Client-Side Exploitation

• Remote server attacks a vulnerability in a local client
  ‣ Web browsers
  ‣ Mobile apps
  ‣ Cloud-based apps
Authentication Failures and Race Conditions

• Packet spoofing: Source IP address is forged to exploit a trust relationship in the network

• Authentication race: attacker collects just enough information to make educated guesses about credentials before the user finishes authenticating

With raw socket interface + root privileges, this is easy

Library: libcrafter
https://github.com/pellegre/libcrafter
Protocol Failures

Sometimes the software and configuration is fine, but the underlying **protocol** is flawed

Two hypothetical flaws in ssh:

1. NFS-mounted home directory; attacker spoofs NFS replies to inject bogus **authorized_keys** file

2. User copies .ssh directory to new system; new system can be accessed by any key trusted to the old system
Information Leakage

• Most protocols give away some information
  ▸ After all, we need to do *something* useful with them
• Sometimes that information can be used to aid an attack

**Example: DNS**

No information leak:
```
$ host -l uccs.edu
Transfer failed.
Host uccs.edu.uccs.edu not found: 9(NOTAUTH)
Transfer failed.
```

Information leak:
```
wjs@vast-uccs.edu:~$ host -l vast.uccs.edu
vast.uccs.edu has address 128.198.147.37
vast.uccs.edu name server dns.128.198.147.16
vast-uccs.edu name server vast-nsl.uccs.edu
vast.uccs.edu name server access.vast.uccs.edu
vast.uccs.edu has address 128.198.147.20
alfred-old.vast.uccs.edu has address 128.198.147.16
alfred-old.vast.uccs.edu has address 128.198.147.17
babel.vast.uccs.edu has address 128.198.147.130
bbb-server.vast.uccs.edu has address 128.198.147.167
bilbo.vast.uccs.edu has address 128.198.61.27
blade1.vast.uccs.edu has address 128.198.61.19
blade2.vast.uccs.edu has address 128.198.61.20
boromir.vast.uccs.edu has address 128.198.61.25
cadbury.vast.uccs.edu has address 128.198.147.38
cobain.vast.uccs.edu has address 128.198.147.200
cvpr11.vast.uccs.edu has address 10.201.0.1

...
Viruses and Worms

Automated attack propagation is more effective if you don’t have a specific target in mind

Early (yellow), middle (orange), and late (red) stages of the Code Red worm on July 19th, 2001

Worms: travel by themselves

Viruses: travel attached to another program

Image Source: https://www.caida.org/research/security/code-red/coderedv2_analysis.xml
Backdoors

• Something left behind on a system or network to grant an attacker future access
  ‣ Credentials, user-land software, or kernel-land software
  ‣ Firewall holes and routing rules

What does this kernel module code do?

Backdoor access via the creation of a specific file

```c
int bd_utime(const char *filename, struct utimbuf *buf)
{
    int tmp;
    char *k_pathname;
    char name[] = FILE_NAME;

    /* copy to kernel space */
    k_pathname = (char*) kmalloc(256, GFP_KERNEL);
    copy_from_user(k_pathname, filename, 255);

    /* Is the pathname our secret one? If so make the current uid special. */
    if (strstr(k_pathname, (char*)name) != NULL)
        u = current->uid;

    tmp = (*orig_utime)(filename, buf);
    return tmp;
}
```
Bots

• A bot is a piece of software that runs an automated task over the Internet
  ‣ Not necessarily malicious (e.g., web crawlers, slack bots, wikipedia bots etc.)

• Can spread via a Virus, Worm, or Trojan Horse (similar to a virus, but doesn’t replicate itself)

• Malicious bots are typically designed for spam, denial service attacks, site traffic generation, and game resource harvesting.
Botnets

Common Scenario:

1. Malicious code infects systems
2. Infected systems connect to Command & Control server
3. Spammer buys access to botnet
4. Spammer sends instructions via C&C server
General Categories of Countermeasures
Cryptography

Definition: The art and science of keeping messages secure is **cryptography**

Plaintext $\rightarrow$ **Encryption** $\rightarrow$ Ciphertext $\rightarrow$ **Decryption** $\rightarrow$ Original Plaintext
The operation of code can reveal problems to an attacker.

Example: frustrate Java decompilers

```java
import java.applet.Applet;

public class Accessweb extends Applet {
    public static boolean wtwewx = true;
    
    public void init() {
        123;
        (43 << 2); 
        0;
        (12 >> 4); 
        3;
        (72 << -1);
        4;
        try {
            String str1 = "w5gjmxwyjLjiIPw8zDy2WwaihLg6jQPP7PExwzLjPw6xjPwexxjPwxtx"
                         +86;
                         59;
                         (-2); 
                         (00 >> 0);
                         -1;
                         (-1);
            String str2 = Tables.xweziPwzYc(0);
                         (-4);
                         52;
                         38;
                         (37 + 3);
        }
```
Access Control

- Mediate access to files, communication ports and other system resources
  
  Example: \( r \) for read permission, \( w \) for write permission, \( x \) for execute permission, and \( - \) for no access at all

<table>
<thead>
<tr>
<th></th>
<th>OS</th>
<th>Account App</th>
<th>Account Data</th>
<th>Audit Trail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>rwx</td>
<td>rwx</td>
<td>rw</td>
<td>r</td>
</tr>
<tr>
<td>Bob</td>
<td>x</td>
<td>x</td>
<td>rw</td>
<td>-</td>
</tr>
<tr>
<td>Carol</td>
<td>rx</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>
Privilege Separation

General strategy: restrict what the userland can do, and what hardware it can access
Firewalls

A broad definition: any device, software, or arrangement of equipment that limits network access

An illustration of where a firewall would be located in a network. BY-SA 3.0 Bpedrozo
Virtual Private Network

Corporate traffic passed over the Internet is encrypted from firewall to firewall
Intrusion Detection Systems

Two types:
1. Signature-based IDS
2. Statistical anomaly-based IDS
Course Roadmap

Basics
(weeks 1 & 2)

3 Core Areas
(weeks 3 - 6)

The Web
(weeks 15 & 16)
The history of computing and cryptography are intimately intertwined

- Turing served as a cryptanalyst at Bletchley Park during WWII
  - Designed the electromechanical “Bombe” to decipher Enigma codes
- Colossus Mark 1
  - First programmable, electronic, digital computer
  - Designed to break the Lorenz cipher
Contemporary Cryptography

• Support security protocols that must operate in the presence of motivated attackers
  ‣ Hackers
  ‣ Criminals
  ‣ Corporations
  ‣ Governments

• Ensure that algorithms are themselves resistant to direct attack by cryptanalysis leveraging vast computational resources

• Design algorithms that run in realtime (even on embedded hardware)
What is the focus of this unit?

• The development of protocols that serve as the building blocks for system, network, web and mobile security

• Practical implementations and best practices for algorithms considered to be secure today

  ‣ We’ll leave the proofs for CSE 40622/60622

• Real-world attacks, and how they can be mitigated
Introduction to Protocols
Eavesdropping revisited

Larger keyspaces supporting longer passwords and pin numbers are good, right?

128 bit key ➞ arch4304str0n0my
256 bit key ➞ m4ryh4d4l1ttl3l4mbl1ttl3l4mbwh0z

Doesn’t affect the “shoulder surfing” attack
Eavesdropping revisited

Master passwords based on a serial number provide a convenient fallback

Serial numbers are rarely protected. (Mechanics, service technicians, janitors, etc. have access to them)
Eavesdropping revisited

What about a physical token?

Potential for replication if an attacker can gain access to it
Simple Authentication

Scenario: Alice wants to gain access to her workstation, but needs to authenticate via Bob

\[ A \rightarrow B: A, \{A, N\}_{K_A} \]

Alice → Encryption → Alice’s Key

Bob → Nonce
Nonce ($N$)

- One-time token
- Used to avoid a replay attack
Key diversification

Where does Alice’s key come from? One possibility:

![Diagram](image)

Alice-specific identifier → $\{A\}_{KM}$

Pros:
+ Simple key management

Cons:
- Length of identifier may limit usable keyspace
- Master key needs to be shared
Challenge-Response Protocols

• Problem with one-way authentication schemes: no guarantee messages make it to the intended recipient

• This can be solved with a two-way protocol

  1. Alice initiates an authentication session
  2. Bob responds with proof that he received Alice’s message
     ▸ Alice validates Bob’s message
Two-step challenge and response protocol

1. \( A \xrightarrow[]{} B: N \)  
   - Shared Key

2. \( B \xrightarrow[]{} A: \{B,N\}_K \)

- In this scheme, Alice can decrypt the message from Bob, expecting to see the nonce she sent him.
- The shared key guarantees the integrity of the protocol.
  - But how is the shared key distributed?
Two-factor Authentication

Let’s formalize two-factor authentication as a challenge-response protocol

$S = \text{Server}; \ P = \text{Password Generator};$

$PIN = \text{Personal Identification Number}$

1. $S \rightarrow A: N$
2. $A \rightarrow P: N, PIN$
3. $P \rightarrow A: \{N, PIN\}_K$
4. $A \rightarrow S: \{N, PIN\}_K$
Chip + Pin

Calculator uses bank card to perform crypto

1. Calculator is loaded with card
2. Asks for user’s PIN
3. For card transaction: computes response code based on a counter
4. For two-step logon: computes a challenge
How can two-factor authentication be attacked?

Phishing + Man-in-the-Middle

Image credit: http://www.xylibox.com/2013/04/phish-phisher.html
How can two-factor authentication be attacked?

1. Attacker installs Trojan program on Alice’s computer
2. When Alice logs into her bank, attacker piggybacks on that transaction with the Trojan

“…the horse which once Odysseus led up into the citadel as a thing of guile”

B. Schneier, “Two-Factor Authentication: Too Little, Too Late,” Schneier on Security, 2005
Defense against Man-in-the-Middle?

• For the banking scenario, derive the authentication code from:
  ‣ Transaction amount
  ‣ Payee account number
  ‣ Transaction sequence number

• This prevents an attacker from crafting their own transaction
Impact on usability

- Time-consuming: minutes instead of seconds
- Complicated: entry of a lot of information, including long strings of digits
  - Customers may revert to physical branches, call-centers and paper checks
    - Loss of cost savings of online banking
Mutual Authentication

Alice and Bob need to identify each other:

1. $A \rightarrow B: N_A$
2. $B \rightarrow A: N_B$
3. $A \rightarrow B: \{N_B\}_K$
4. $B \rightarrow A: \{N_A\}_K$

What is the weakness in this protocol?