50.020 Security
Lecture 7: Web Security II

Midterm Exam: some information

- Duration: 1.5 hour
- Coverage:
  - Week 1- Week 5 if the exam takes place in Week 6
  - Week 1- Week 6 if the exam take place in Week 8.
- Timing:
  - Option 1 (as scheduled before): Week 6 lab time
  - Option 2 (if you think necessary to postpone): Week 8 Monday 1:15-3pm (please come 15 mins earlier than usual Monday class; exam will start at 1:20pm, and ends at 2:50 in case you need to go for next lesson)
  - Which option you prefer to? Please get back to me by 23:59 this Wed (19 Feb). Big thanks to the coordinator for your time/effort on creating the survey.
  - Let's confirm the date this Thursday.

Overview

Discussed in last lecture:

- Most common vulnerabilities: OWASP
  - SQL injection
  - Cross-site scripting (XSS)
  - Cross-Site Request Forgery (CSRF)
  - Weak authentication

This lecture:

- Command Injection
  - Shellshock use case
  - Reverse Shell
- DoS attacks on server
- Mirai/ IoT security
- Cloud security (brief)

Command Injection

Shellshock: Introduction

- "Shellshock" is the name given to a Bash vulnerability discovered late 2014
  - One of the first "named" exploits with much media attention
- Bash is one of the most common *command language interpreters*
- Bash had vulnerabilities in its environmental variable parser
- These can be used to *execute commands* on a target machine
- Much media hype, damage unclear

SUTD Singapore University of Technology & Design
4 hrs ·

In this era of technology, cyber security is of paramount importance. This is why "Security and Communication" is one of the tracks that our undergraduates can pursue in their course of study in the Information Systems Technology and Design (ISTD) Pillar.

Read more information about ISTD here:
http://istd.sutd.edu.sg/about/

Hackers Are Already Using the Shellshock Bug to Launch Botnet Attacks | WIRED

With a bug as dangerous as the "shellshock" security vulnerability discovered yesterday, it takes less than 24 hours to go from proof-of-concept to pandemic.

WIRED.COM | BY ANDY GREENBERG

Example of media coverage

Shellshock: Details on disclosure

- Bug was discovered by Stephane Chazelas
- *Responsible disclosure* was used, only Bash maintainer and major distributions were notified
- The bug was fixed and a coordinated public disclosure was coordinated:

`http://www.openwall.com/lists/oss-security/2014/09/24/11`

- Most people had their system patched before it was even in the mainstream media.
- A CVE number was assigned to coordinate the efforts:

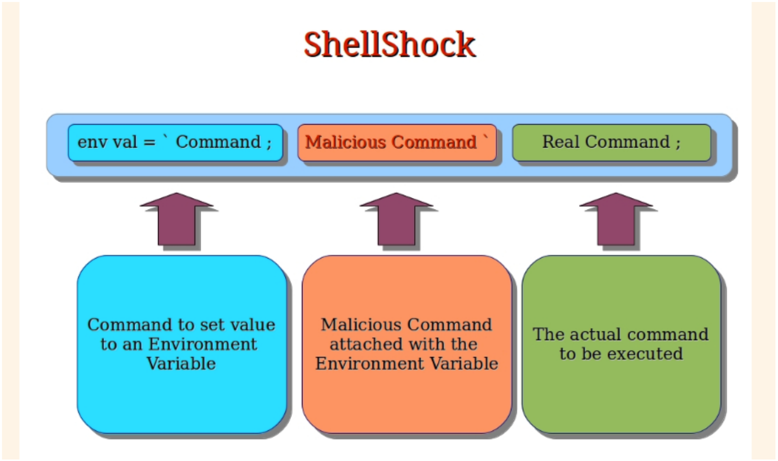
`https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271`

Shellshock: Who could be attacked?

- Basically every Linux (before patched) had this bug. Mac too (before pathced).
- Requires the victim to execute Bash programs with env variables set by attacker
- cPanel (web hosting interface) is vulnerable, running on 2.9% of sites[1]

[1] `http://blog.sucuri.net/2014/09/bash-vulnerability-shell-shock-thousands-of-cpanel-sites-are-hig`
`html`

Shellshock: How Does Shellshock work?


ShellShock
env val = ` Command ;
Malicious Command `
Real Command ;
Command to set value to an Environment Variable
Malicious Command attached with the Environment Variable
The actual command to be executed

Shellshock: How Does Shellshock work?

- Bash will parse environmental variables when started
- First bug found: env variables with `() { code; }` syntax can be used to export functions
  - Spawning bash will parse the env variables
  - They will actually execute commands following the `(){:;}` part!
  - So env variable `() { :; } echo foo` will cause echo to be executed
  - Test if machine has the bug: (should not print `vulnerable`)

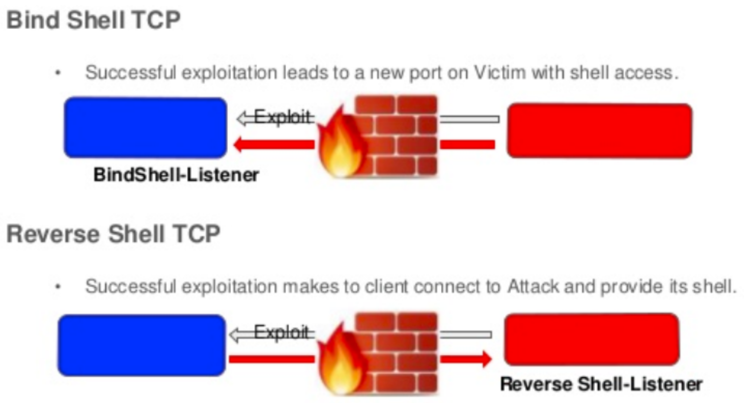
```
env x='() { :;}; echo vulnerable' bash -c 'echo foo'
```

Shellshock: What can be done on victim systems

- Ping back to some machine (benign)
- Open reverse shell (malicious)
  - `http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet`
- Download botnet client, join botnet:
  - `http://pastebin.com/mG1grQwK`

Reverse Shell: when needed?

A reverse shell is an attack technique used when the target machine is not directly reachable (due to firewall, NAT, etc).

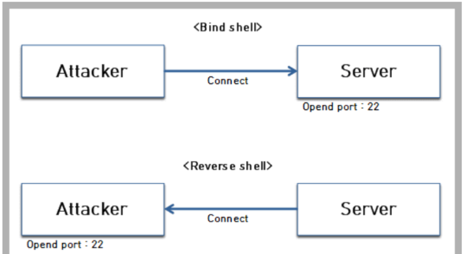

Bind Shell TCP
• Successful exploitation leads to a new port on Victim with shell access.
Exploit
BindShell-Listener
Reverse Shell TCP
• Successful exploitation makes to client connect to Attack and provide its shell.
Exploit
Reverse Shell-Listener

Reverse Shell: What is it?

- Bind shell : A shell that **the attacker uses after connecting to the server**. A bind shell is **setup on the target host** and binds to a specific port to listen for an incoming connection from the attacker.
- Reverse shell: A shell that **the attacker uses after the server connects to the attacker**. A reverse shell is a shell **initiated from the target host** back to the attacker who is in a listening state to pick up the shell.

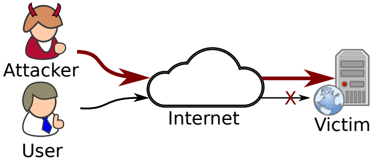
You can open a reverse shell using Netcat (exercise in the lab) or other tools.

Note: Port 22 in the picture below can be any other unfiltered port.



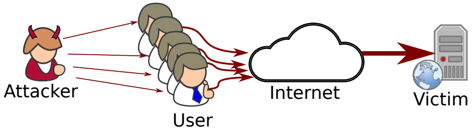
Server security: Denial of service attacks

Denial of service (DoS)



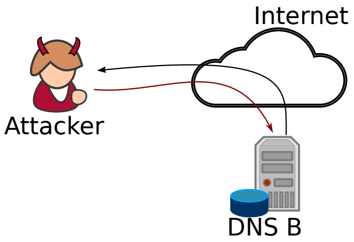
- Any server on the web has limited bandwidth (HW/link)
- *Denial of Service* (DoS) attacks exhaust this bandwidth
- Example: simple DoS use ICMP ping flooding
    - Target receives large number of pings, replies to each

DDoS attacks



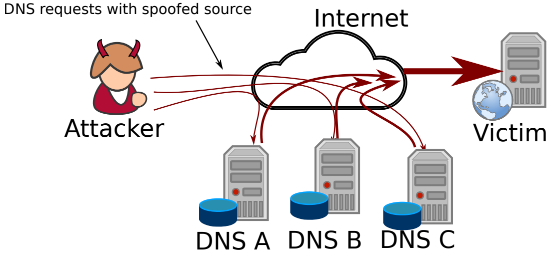
- DoS attacks need high bandwidth for attacker
- *Distributed DoS* allows multiple uplinks
  - Often relies on Botnets or many users (anonymous/LOIC)
- Similar attacks are possible with many other protocols (TCP SYN, DNS, . . .)
  - Ideally, high *amplification* of attacker's effort for the victim

DNS request messages



- Domain name service protocol allows lookup of IP-addresses for names
- Many open DNS servers on the internet who you can query
- Server will respond with cached DNS information
- Result can be bigger than query
- How to abuse for DDoS?

DNS amplification DDos



- Ask for big DNS TXT resource record ($\approx$ 4kB) with 60B message
- Set victim IP as source address
- Open DNS servers will send big reply packets to victim
- Amplification factor: around 66x

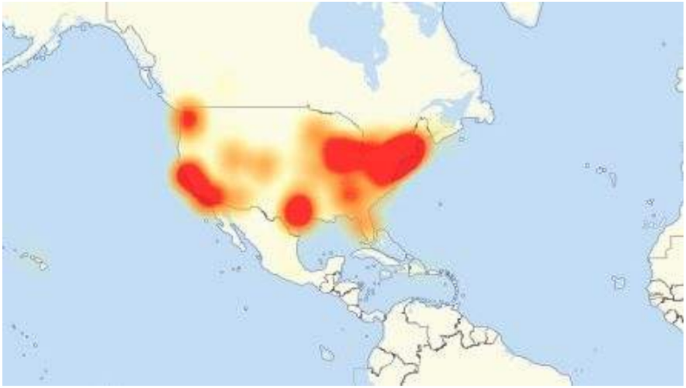
"Live" DDoS attack

`http://www.digitalattackmap.com`

Mirai

Mirai in the media

2. 'Mirai' Took Out Amazon, Spotify, Twitter and More Websites in a DDOS Attack



The Internet was down throughout the country (Down Detector)

The morning of October 21 saw widespread internet outages caused by a massive DDOS

Mirai

- Very widely spread malware that primarily infected IoT Devices
- Devices were then used as botnet to launch massive DoS attacks
    - 620 Gbit/s to 1Tbit/s
    - comparison: SUTD downlink is 1400 mbit/s (1/700th)
- Interesting write-up on suspected author on Kreb's blog
- Mirai's source code was leaked and is available at
    - `https://github.com/jgamblin/Mirai-Source-Code`

Why did Mirai infect victims?

- Initial wave of infections targeted IP cameras by manufacturer XM
  - 400,000 devices were allegedly infected
  - Cameras had open Telnet ports
  - Default password for admin
- Manufacturer blames users for not changing password, and not updating firmware
  - `http://www.xiongmaitech.com/index.php/news/info/12/76` (chinese)
  - But in many cases, the device's storage was read only. . .
- This will likely happen more often for IoT devices in the future
- After successful login, attacker would download and install botnet client

Cloud Security

Cloud Security

- We will not discuss much about cloud security in this class
- The basics:
  - Cloud application run on third party servers
  - Your data will be stored on those servers
  - You need to securely connect to those servers
- Connecting to the cloud securely is easy using TLS
- Only problem remaining: trusting the cloud
  - Trust they do not get compromised
  - Trust they don't sell your data
  - Trust they protect again brute force attacks on authentication etc.

Example: iCloud leaks

- Private (i.e. confidential) pictures of celebreties were published
- Most data seems to have been hosted on iCloud
- Most likely, multiple attackers pooling their findings (collectors)
- iCloud uses Apple ID and password
  - "New account"
  - "Find my iPhone" API did not restrict number of incorrect tries
  - These could be exploited to do password guessing attacks
- Apple now introduced two-factor authentication

Conclusion Web Security

- Server security depends on:
  - Proper access control
  - Proper processing of user-provide input
  - Timely application of security patches
- User device security depends on:
  - Secure execution of downloaded content
    - Virus scanners can help
    - Sandboxing can help
  - Minimizing attack surface
    - Correct configuration of device
    - Personal firewalls
    - NATs and other network measures
  - Application with proper user-provided input sanitization