CIS Control 4 - Secure Configuration of Enterprise Assets and Software

Establish and maintain the secure configuration of enterprise assets (end-user devices, including portable and mobile; network devices; non-computing/IoT devices; and servers) and software (operating systems and applications).

☑	4.1	Establish and Maintain a Secure Configuration Process	Applications	●	●	●
☑	4.2	Establish and Maintain a Secure Configuration Process for Network Infrastructure	Network	●	●	●
☑	4.3	Configure Automatic Session Locking on Enterprise Assets	Users	●	●	●
☑	4.4	Implement and Manage a Firewall on Servers	Devices	●	●	●
☑	4.5	Implement and Manage a Firewall on End-User Devices	Devices	●	●	●
☑	4.6	Securely Manage Enterprise Assets and Software	Network	●	●	●
☑	4.7	Manage Default Accounts on Enterprise Assets and Software	Users	●	●	●
☑	4.8	Uninstall or Disable Unnecessary Services on Enterprise Assets and Software	Devices		●	●
☑	4.9	Configure Trusted DNS Servers on Enterprise Assets	Devices		●	●
☑	4.10	Enforce Automatic Device Lockout on Portable End-User Devices	Devices		●	●
☑	4.11	Enforce Remote Wipe Capability on Portable End-User Devices	Devices		●	●
☑	4.12	Separate Enterprise Workspaces on Mobile End-User Devices	Devices			●

Secure Configuration Management

- Not as important as it was in 2000
- Most attacks do not go after misconfigurations
- Still required by compliance
- Difference between OS and Applications
- Applications MATTER

Secure Configuration Management: Applications

- CIS guides ← Many scanner can scan to a policy
  - Good for enforcement!
- Vendor Documents
- Cloud deployments
- Complex stacks of technology
- Some NAC devices can do this as well

CIS Control 7 - Continuous Vulnerability Management

Develop a plan to continuously assess and track vulnerabilities on all enterprise assets within the enterprise's infrastructure, in order to remediate, and minimize, the window of opportunity for attackers. Monitor public and private industry sources for new threat and vulnerability information.

☑	7.1	Establish and Maintain a Vulnerability Management Process	Applications	●	●	●
☑	7.2	Establish and Maintain a Remediation Process	Applications	●	●	●
☑	7.3	Perform Automated Operating System Patch Management	Applications	●	●	●
☑	7.4	Perform Automated Application Patch Management	Applications	●	●	●
☑	7.5	Perform Automated Vulnerability Scans of Internal Enterprise Assets	Applications		●	●
☑	7.6	Perform Automated Vulnerability Scans of Externally-Exposed Enterprise Assets	Applications		●	●
☑	7.7	Remediate Detected Vulnerabilities	Applications		●	●

Network Device Management

Establish, implement, and actively manage (track, report, correct) network devices, in order to prevent attackers from exploiting vulnerable network services and access points.

☑	12.1	Ensure Network Infrastructure is Up-to-Date	Network	●	●	●
☑	12.2	Establish and Maintain a Secure Network Architecture	Network		●	●
☑	12.3	Securely Manage Network Infrastructure	Network		●	●
☑	12.4	Establish and Maintain Architecture Diagram(s)	Network		●	●
☑	12.5	Centralize Network Authentication, Authorization, and Auditing (AAA)	Network		●	●
☑	12.6	Use of Secure Network Management and Communication Protocols	Network		●	●
☑	12.7	Ensure Remote Devices Utilize a VPN and are Connecting to an Enterprise's AAA Infrastructure	Devices		●	●
☑	12.8	Establish and Maintain Dedicated Computing Resources For all Administrative Work	Devices			●

Network Device Management

- Network devices are often ignored
- Often for years…. Decades?
- They need to be patched just like… Everything
- Huge post exploitation wins for attackers
- Key is change management
- Nmap can help
- Vulnerability management can address 98% of this

Vulnerability Management

Vulnerability Management

- Same as it was 10+ years ago
- Vendors have not changed with the times
- Test and scan for external vulnerabilities
- Some companies are moving towards credentialed scans
- Very little in actual innovation

Vulnerability Prioritization

- New focus on prioritization
- Address the most critical issues first
- While prioritization can be a great approach it can also be a crutch
- Addressing only the High and Critical issues
  - Many attackers will exploit Low and Informational issues
- Very difficult for vendors to do this without organizational and service context

Low and Informational Blind Spots: Example

10.10.10.133 (tcp/23)

```
Here is the banner from the remote Telnet server :

------------------------------ snip ------------------------------
Login:

------------------------------ snip ------------------------------
```

10.10.10.134 (tcp/23)

```
Here is the banner from the remote Telnet server :

------------------------------ snip ------------------------------
Login:

------------------------------ snip ------------------------------
```

10.10.10.135 (tcp/23)

```
Here is the banner from the remote Telnet server :

------------------------------ snip ------------------------------
router>

------------------------------ snip ------------------------------
```

Question:
How Many of Your Organization's Address Low and Informational Issues?

Addressing Vulnerabilities: The Wrong Way

- Many organizations address vulnerabilities by IP address
- For example: 1,000 IP addresses x ~25 vulnerabilities per IP = 25,000 issues to address
- This can be daunting
- Because of this we can see why so many companies focus on prioritization
- However, this approach is almost always wrong

Key Point:
Focus on Grouping Issues by Vulnerability, Not by IP Address

Addressing Vulnerabilities: The Correct Way

- Stop focusing on IP addresses and ranges
- Focus on the vulnerabilities
- Instead of 25,000 total vulnerabilities you will be dealing with a few hundred that repeat on multiple systems
- Use automation and address them as groups of issues
- This approach works regardless of the tool you use
- Consider it an “Open Source Technique”
- With this method IANS faculty have addressed over 1 million IP address, all vulnerabilities in less than 3 weeks

MITRE ATT&CK

Enterprise Matrix

Below are the tactics and technique representing the MITRE ATT&CK Matrix™ for Enterprise. The Matrix contains information for the following platforms: Windows, macOS, Linux, AWS, GCP, Azure, Azure AD, Office 365, SaaS.

Last Modified: 2019-10-09 18:48:31.906000

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Application Access Token	Bash History	Application Window Discovery	Application Access Token	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Application Deployment Software	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Addition	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bit			ect Model d COM	Data from Cloud Storage Object	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Bypass U Co			Remote	Data from Information Repositories	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	Clear Com			shishing	Data from Local System	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Cl			pts	Data from Network Shared Drive	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Code			lash	Data from Removable Media	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compile A			icket	Data Staged	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Compile			Protocol	Email Collection	Fallback Channels	Transfer Data to Cloud Account	Network Denial of Service
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Compo			Copy	Input Capture	Multi-hop Proxy		Resource Hijacking
	InstallUtil	Component Firmware	Extra Window Memory Injection	Component Object Model Hijacking	Input Capture	Peripheral Device Discovery	Remote Services	Man in the Browser	Multi-Stage Channels		Runtime Data Manipulation
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Connection Proxy	Input Prompt	Permission Groups Discovery	Replication Through Removable Media	Screen Capture	Multiband Communication		Service Stop

Exploit Public-Facing Application

External Remote Services

CIS Control 16 - Application Software Security

Manage the security life cycle of in-house developed, hosted, or acquired software to prevent, detect, and remediate security weaknesses before they can impact the enterprise.

☑	16.1	Establish and Maintain a Secure Application Development Process	Applications	●	●
☑	16.2	Establish and Maintain a Process to Accept and Address Software Vulnerabilities	Applications	●	●
☑	16.3	Perform Root Cause Analysis on Security Vulnerabilities	Applications	●	●
☑	16.4	Establish and Manage an Inventory of Third-Party Software Components	Applications	●	●
☑	16.5	Use Up-to-Date and Trusted Third-Party Software Components	Applications	●	●
☑	16.6	Establish and Maintain a Severity Rating System and Process for Application Vulnerabilities	Applications	●	●

Right of Boom

☑	16.7	Use Standard Hardening Configuration Templates for Application Infrastructure	Applications	●	●
☑	16.8	Separate Production and Non-Production Systems	Applications	●	●
☑	16.9	Train Developers in Application Security Concepts and Secure Coding	Applications	●	●
☑	16.10	Apply Secure Design Principles in Application Architectures	Applications	●	●
☑	16.11	Leverage Vetted Modules or Services for Application Security Components	Applications	●	●
☑	16.12	Implement Code-Level Security Checks	Applications		●
☑	16.13	Conduct Application Penetration Testing	Applications		●
☑	16.14	Conduct Threat Modeling	Applications		●

Right of Boom

Executive Problem Statement

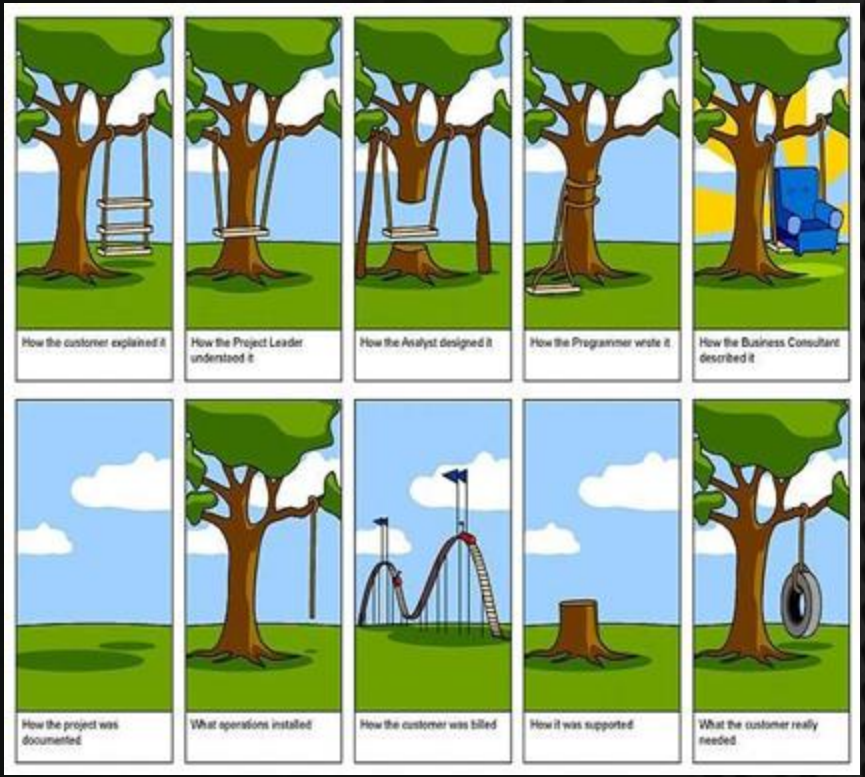
Basic Questions:

- How can we quickly secure our aps?
- Training is very expensive
- Tools can be very expensive
- Changing all processes to incorporate security takes a lot of time

A helpful image of what an “executive” may look like

Software Development Lifecycle

- Continuous builds
- Continuous improvement
- Security is often bolted on on the end
- This is expensive
- This is also dangerous
- Security testing is something that should be done throughout the process
- Beginning, throughout, and end



But Security is Hard

- Not really
- In fact, most security testers know less about development than you do
- Different skill set
- It is easier to teach a web developer security, than it is to teach a tester development
- Lots of free tools and tricks
- 80/20 rule

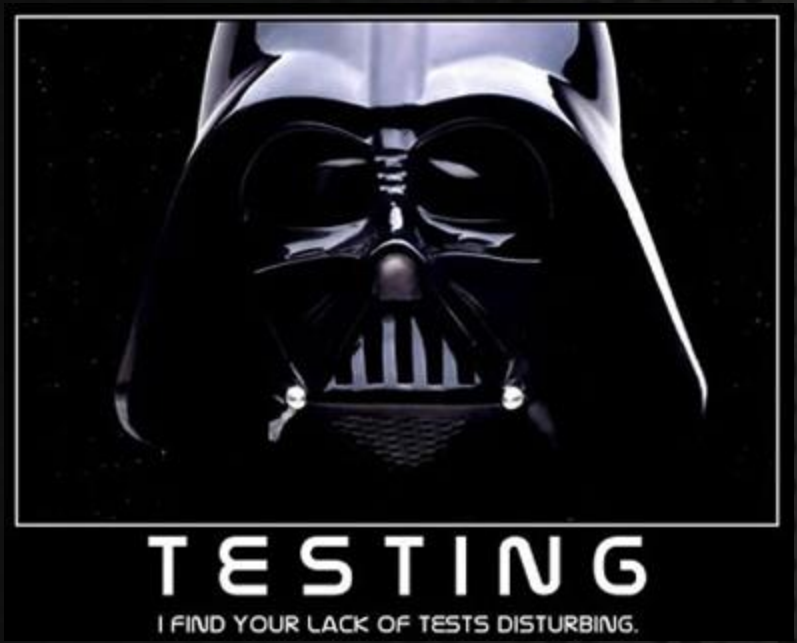
Where and When to Test

- Many of the tools we will talk about are so easy they should be used every build cycle.
- That is, nightly if possible
- Weekly at a minimum
- BHIS recommends a different member of the team test, review and address the issues each time
- Test everything, the tools are so easy to use there is no good reason not to
- Believe it or not, it will make you a better developer

Testing never seem to end.
It just goes on and on my friends!
Kevin, started hacking and not knowing what it was..
Now he'll just keep on hacking it forever
Just because..

What to Test For?

- Things which can be easily detected with an automate tool
- Cross Site Scripting
- SQL Injection
- Command Injection
- Misconfigurations
- The above attacks represent roughly 80 - 85% of the vulnerabilities bad guys attack



Do the Tools Cover Everything?

- No
- Automated tools do a great job
- But they miss
- Logic errors
- Permission errors
- Stored Cross Site Scripting
- Cross Site Request Forgery
- These vulnerabilities require manual testing

Self Test Before the Test

- Web penetration testers do not want to find and report on 100's of XSS vulnerabilities
- The good ones don't anyway
- Best thing to do before you get a test?
- Run these tools and share the results with the tester
- It will greatly speed up the test
- It will allow the tester to focus on harder issues like business logic errors
- Better value for you and the tester will provide a far better report
- Plus, self-assessment should happen on a regular basis

Tools, Tools, Tools

- Burp Pro – Not free, but cheap and awesome
- W3AF – Automatic web security scanner
- $0.00
- Zed Attack Proxy – ZAP
- -$0.00
- Nikto – Free web scanner
- These tools are better than most tools which cost $20K or more
- If you know how to use them

Burp

- Easily the most heavily used tool by most web testers
- It does not have a cool GUI
- There is no place to insert a URL and have it scan
- Very cool
- Worth every penny
- You have to set it up as a proxy, then choose what you scan
- Google “Configure proxy for <INSERT BROWSER HERE>
- Set your proxy settings to 127.0.0.1 8080

Intercept Mode

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Options

Intercept | History | Options

Request to http://testphp.vulnweb.com:80 [176.28.50.165]

Forward | Drop | Intercept is on | Action

Raw | Headers | Hex

```
GET / HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:24.0) Gecko/20100101 Firefox/24.0
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
Connection: keep-alive
```

Scope

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Options

Intercept | History | Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status
1	http://testphp.vulnweb.com	GET	/			200
4	http://testphp.vulnweb.com	GET	/Flash/add.swf			200

http://testphp.vulnweb.com/Flash/add.swf

- Add to scope
- Spider from here
- Do an active scan
- Do a passive scan
- Send to Intruder Meta+I
- Send to Repeater Meta+R
- Send to Sequencer
- Send to Comparer (request)
- Send to Comparer (response)
- Show response in browser
- Request in browser ▸
- Engagement tools ▸
- Show new history window
- Add comment
- Highlight ▸
- Delete item
- Clear history
- Copy URL
- Copy as curl command
- Copy links
- Save item
- Proxy history help

Crawl

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer

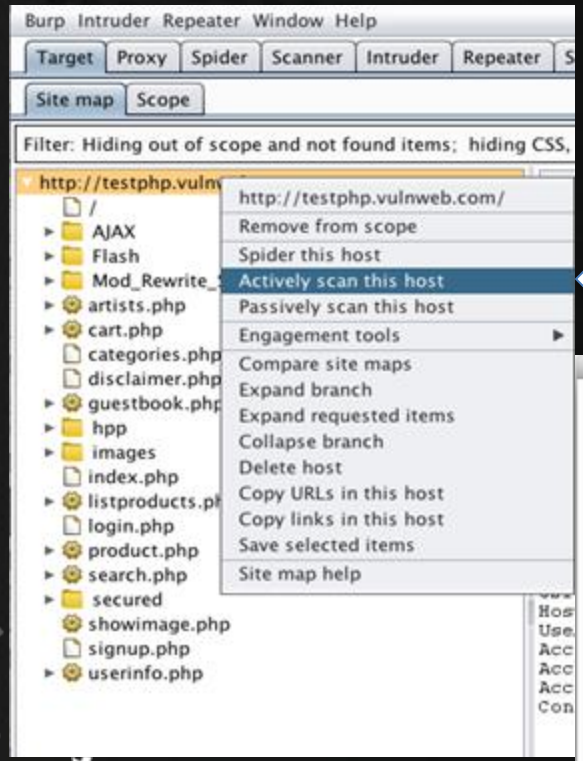
Site map | Scope

Filter: Hiding out of scope and not found items; hiding CSS, image and

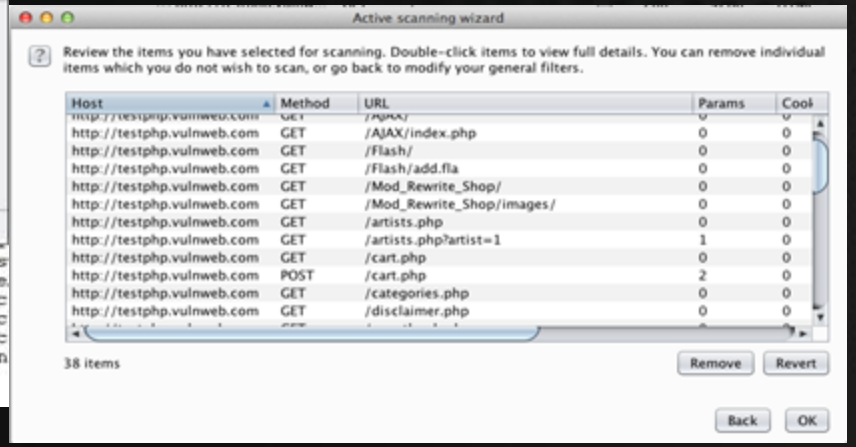
http://testphp.vulnweb.com

- http://testphp.vulnweb.com/
- Add to scope
- Spider this host
- Actively scan this host
- Passively scan this host
- Engagement tools ▶
- Compare site maps
- Expand branch
- Expand requested items
- Delete host
- Copy URLs in this host
- Copy links in this host
- Save selected items
- Site map help

Active Scan



Don’t Forget to Verify the Scope!!



Scan Running

Burp Suite Professional v1.5.18 – licensed to BHIS [5 user license]

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Options | Alerts

Results | Scan queue | Live scanning | Options

#	Host	URL	Status	Issues	Requests	Errors	Insertion p
16	http://testphp.vulnweb.com	/userinfo.php	50% complete	2	128		7
18	http://testphp.vulnweb.com	/Flash/add.fla	25% complete	1	22		3
24	http://testphp.vulnweb.com	/hpp/params.php	28% complete	3	96		6
25	http://testphp.vulnweb.com	/hpp/params.php	25% complete	3	94		7
26	http://testphp.vulnweb.com	/search.php	60% complete	2	66		4
27	http://testphp.vulnweb.com	/search.php	33% complete	3	63		5
28	http://testphp.vulnweb.com	/search.php	11% complete	2	44		8
29	http://testphp.vulnweb.com	/showimage.php	20% complete		15		4
30	http://testphp.vulnweb.com	/listproducts.php	20% complete	2	14		4
31	http://testphp.vulnweb.com	/listproducts.php	16% complete	3	11		5
32	http://testphp.vulnweb.com	/listproducts.php	waiting				
33	http://testphp.vulnweb.com	/signup.php	waiting				
34	http://testphp.vulnweb.com	/product.php	waiting				
35	http://testphp.vulnweb.com	/product.php	waiting				
36	http://testphp.vulnweb.com	/secured/	waiting				
37	http://testphp.vulnweb.com	/secured/newuser.php	waiting				
38	http://testphp.vulnweb.com	/secured/newuser.php	waiting				

Results

Burp Suite Professional v1.5.18 - licensed to BHIS [5 user license]

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Options | Alerts

Results | Scan queue | Live scanning | Options

- http://safebrowsing.clients.google.com
- http://testphp.vulnweb.com

- Cross-site scripting (reflected) [6]
- Flash cross-domain policy
- Cleartext submission of password [2]
- SQL injection [6]
- Password field with autocomplete enabled [2]
- Cross-domain Referer leakage [5]
- Email addresses disclosed [12]
- HTML does not specify charset [6]
- Frameable response (potential Clickjacking) [20]
- Content type incorrectly stated [2]
- Directory listing [3]

Advisory | Request | Response

Email addresses disclosed

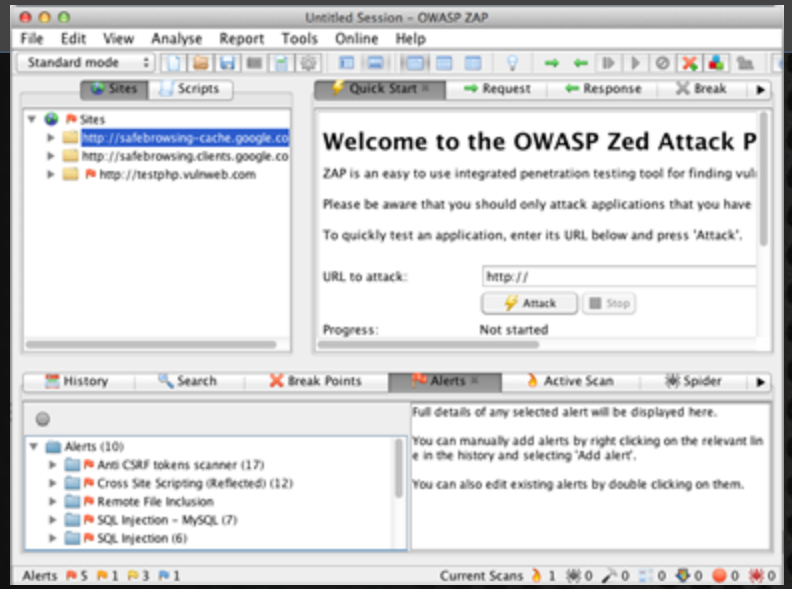
Issue: **Email addresses disclosed**
Severity: **Information**
Confidence: **Certain**
Host: **http://testphp.vulnweb.com**
Path: **/**

Issue detail

The following email address was disclosed in the response:

ZAP!

- Free from OWASP
- Setup is similar to Burp
- Free
- Strong Development Core
- Free
- Has the ability to intercept and modify requests
- Free
- Has the ability to do automated scanning
- Did we mention it was free?
- https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project



Bypasses never seem to end.
They just go on and on my friends!
SubTee, started hacking and not knowing what it was..
Now we will just keep on hacking it forever
Just because..

Getting Caught

Client malware detection and countermeasures			
HTTP viewstate covert channel - VSAgent; Port 443	2/1/2018 9:33	blocked	required authenticated proxy which is not compiled into client agent
DNSCat C2 channel; Port 53	2/1/2018 9:37	blocked	McAfee signature fired, and deleted malware
Metasploit HTTPS Meterpreter Shell code injected into memory via PowerShell; Port 443	1/31/2018 15:30	blocked	script would not seem to execute. No shell connection received
Metasploit TCP Meterpreter Shell code injected into memory via PowerShell (obfuscated with Unicorn); Port 443	2/1/2018 9:35	blocked	McAfee signature fired, and deleted malware
PowerShell Empire PowerShell code injected into memory; Port 443	2/1/2018 9:48	allowed	Command shell active
Raw malware EXE - Metasploit; Port 443; templated using write.exe	2/1/2018 9:56	allowed	Command shell active
Encoded malware EXE - Metasploit; Port 443; templated using write.exe	2/1/2018 9:57	allowed	Command shell active
MS-Office Document malicious macro; HTTPS port 443	2/1/2018 14:28	allowed	Command shell active
MS-Office Document malicious macro; TCP Port 8080	2/1/2018 14:34	blocked	McAffee Detected Malware
Cleartext communication with Netcat tool; Port 8443	2/1/2018 10:00	allowed	Anything that communicates with a TLS port such as 443 or 8443 is allowed through the perimeter without inspection
Metasploit Reverse TCP single stage EXE file.	2/1/2018 14:40	allowed	Command shell active
Metasplot Reverse TCP single stage Visual Basic file.	2/1/2018 14:39	blocked	McAffee Detected Malware
ICMP C2 Channel	2/1/2018 10:52	allowed	ICMP command shell established

Getting Caught 2

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
Accessibility Features	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	Application Deployment Software	Command-Line Interface	Audio Capture	Automated Exfiltration	Commonly Used Port
AppCert DLLs	Accessibility Features	Binary Padding	Brute Force	Application Window Discovery	Distributed Component Object Model	Dynamic Data Exchange	Automated Collection	Data Compressed	Communication Through Removable Media
AppInit DLLs	AppCert DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Vulnerability	Execution through API	Browser Extensions	Data Encrypted	Connection Proxy
Application Shimming	AppInit DLLs	Code Signing	Credentials in Files	Network Service Scanning	Logon Scripts	Execution through Module Load	Clipboard Data	Data Transfer Size Limits	Custom Command and Control Protocol
Authentication Package	Application Shimming	Component Firmware	Exploitation of Vulnerability	Network Share Discovery	Pass the Hash	Graphical User Interface	Data Staged	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Bootkit	Bypass User Account Control	Component Object Model Hijacking	Forced Authentication	Peripheral Device Discovery	Pass the Ticket	InstallUtil	Data from Local System	Exfiltration Over Command and Control Channel	Data Encoding
Browser Extensions	DLL Search Order Hijacking	DLL Search Order Hijacking	Hooking	Permission Groups Discovery	Remote Desktop Protocol	LSASS Driver	Data from Network Shared Drive	Exfiltration Over Other Network Medium	Data Obfuscation
Change Default File Association	Exploitation of Vulnerability	DLL Side-Loading	Input Capture	Process Discovery	Remote File Copy	Mshta	Data from Removable Media	Exfiltration Over Physical Medium	Domain Fronting
Component Firmware	Extra Window Memory Injection	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning	Query Registry	Remote Services	PowerShell	Email Collection	Scheduled Transfer	Fallback Channels
Component Object Model Hijacking	File System Permissions Weakness	Disabling Security Tools	Network Sniffing	Remote System Discovery	Replication Through Removable Media	Regsvcs/Regasm	Input Capture		Multi-Stage Channels
Create Account	Hooking	Exploitation of Vulnerability	Password Filter DLL	Security Software Discovery	Shared Webroot	Regsvr32	Man in the Browser		Multi-hop Proxy
DLL Search Order Hijacking	Image File Execution Options Injection	Extra Window Memory Injection	Private Keys	System Information Discovery	Taint Shared Content	Rundll32	Screen Capture		Multiband Communication
External Remote Services	New Service	File Deletion	Replication Through Removable Media	System Network Configuration Discovery	Third-party Software	Scheduled Task	Video Capture		Multilayer Encryption
File System Permissions Weakness	Path Interception	File System Logical Offsets	Two-Factor Authentication Interception	System Network Connections Discovery	Windows Admin Shares	Scripting			Remote File Copy

Key Takeaways

- Moving from “Can we be hacked?”
  - To..
- “What can we detect?”
- We (finally) have a framework for this with MITRE
- We also have a large number of tools in their infancy to help automate this
- Start by finding gaps. Fill them. Move on.
- Start with the framework