COMP6443 : Topic 5 (Week 9)

DevSecOps

A NOTE ON ETHICS / LEGALITY

- UNSW hosting this course is an extremely important step forward.
- We expect a high standard of professionalism from you, meaning:
  - Respect the property of others and the university
  - Always abide by the law and university regulations
  - Be considerate of others to ensure everyone has an equal learning experience
  - Always check that you have written permission before performing a security test on a system

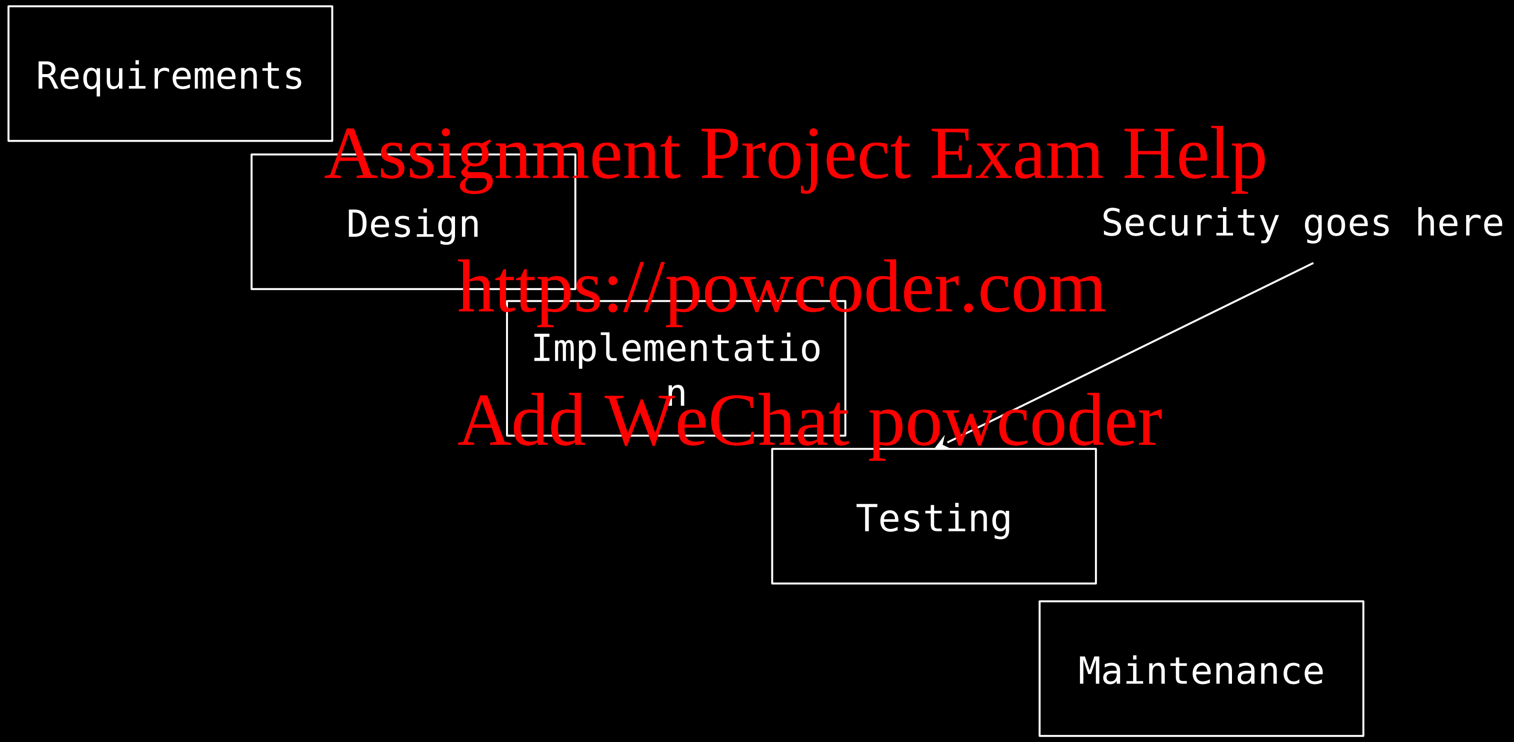
Always err on the side of caution. If you are unsure about

Agile vs. Waterfall

Waterfall development

- Software has been traditionally developed as a sequential project, visualised as a waterfall, with the output of each phase becoming the input to the next.
- Pros:
  - Clear scheduling
  - Task dependency
  - Accurate planning
- Cons:
  - Inflexibility for changing requirements while a project is being executed
  - Schedule blowout if one phase holds up the subsequent phases
  - Integration occurs at the very end of the process

Security in a waterfall model



Requirements

Design

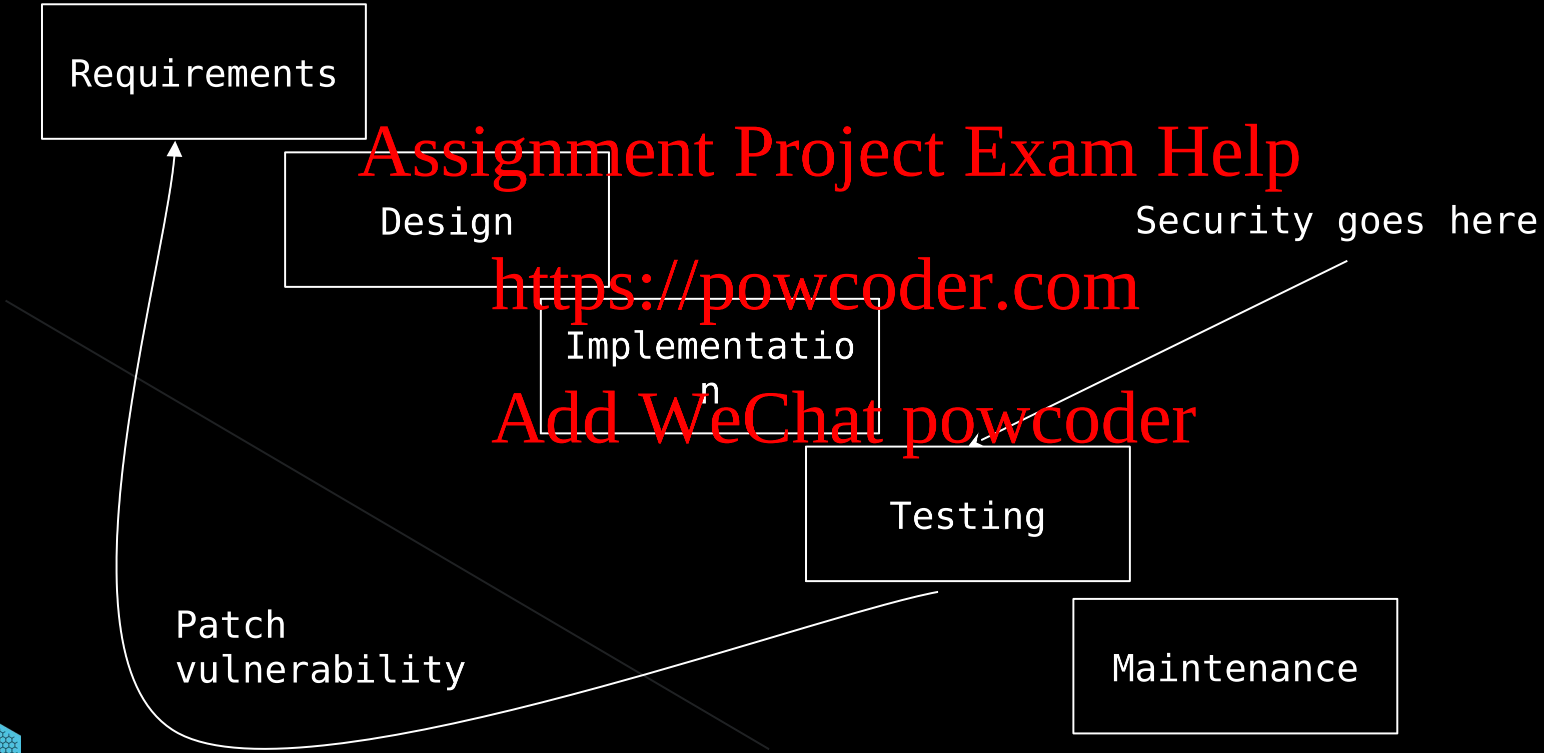
Implementation

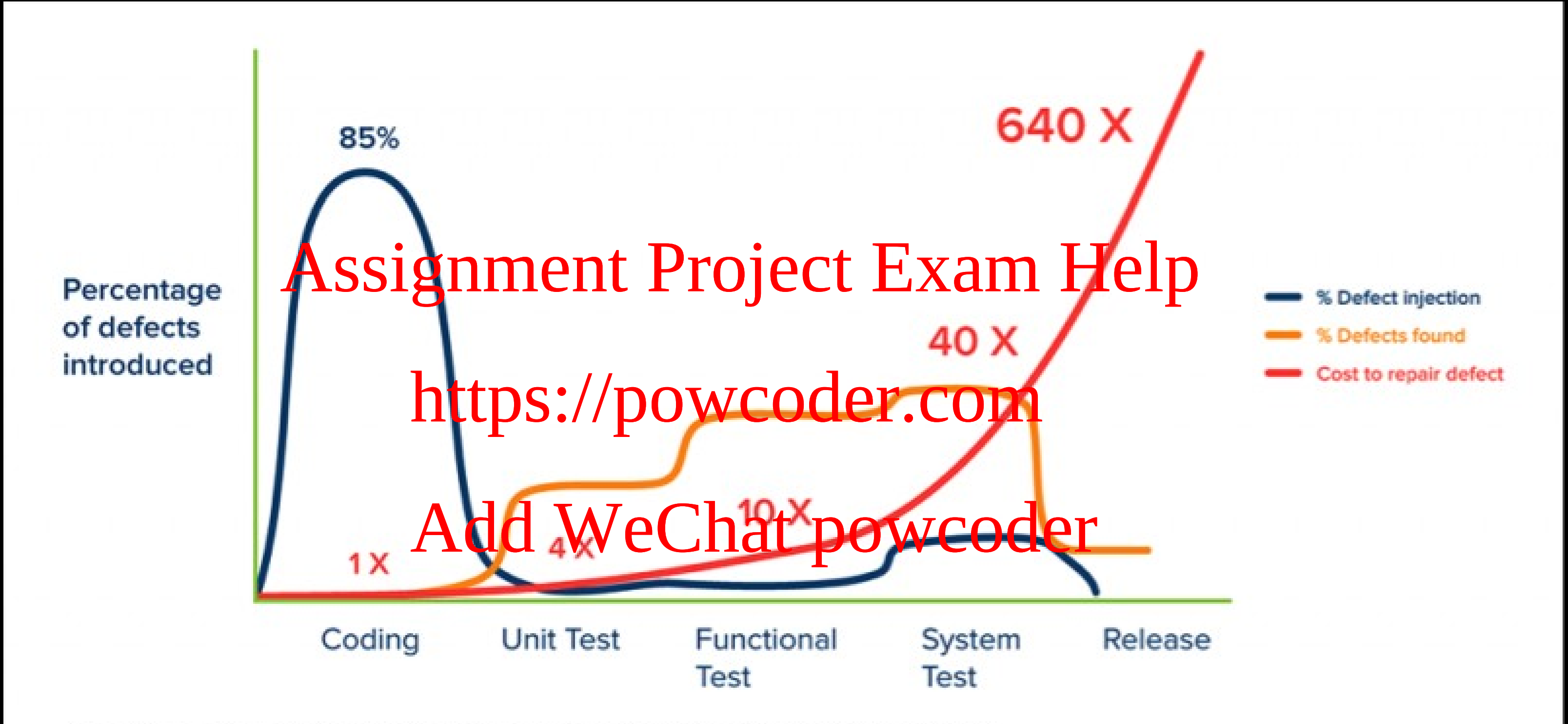
Testing

Maintenance

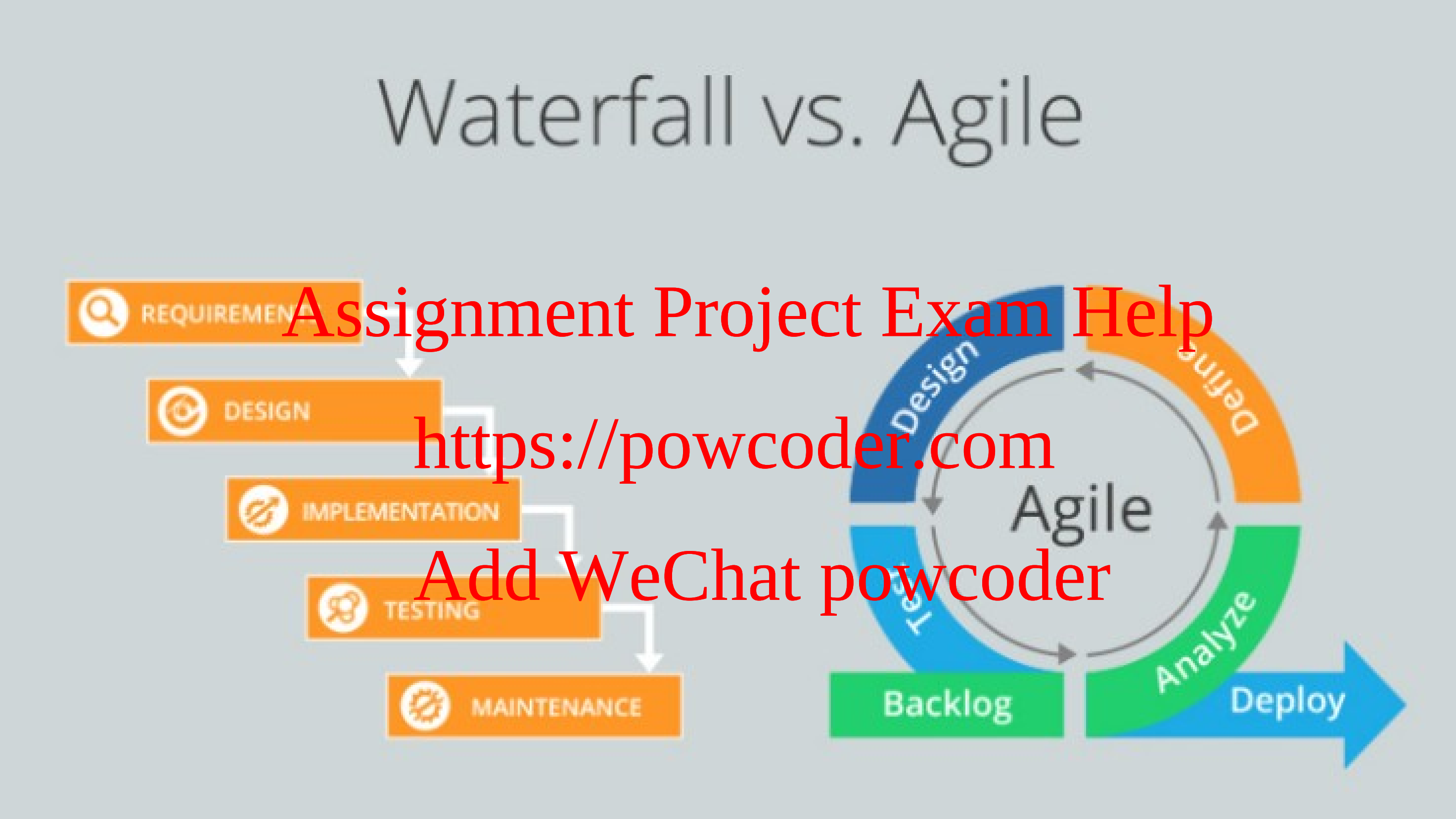
Security goes here

Security in a waterfall model





Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality.*



Source: http://ouriken.com/blog/which-one-is-right-for-you-waterfall-or-agile/

Agile manifesto

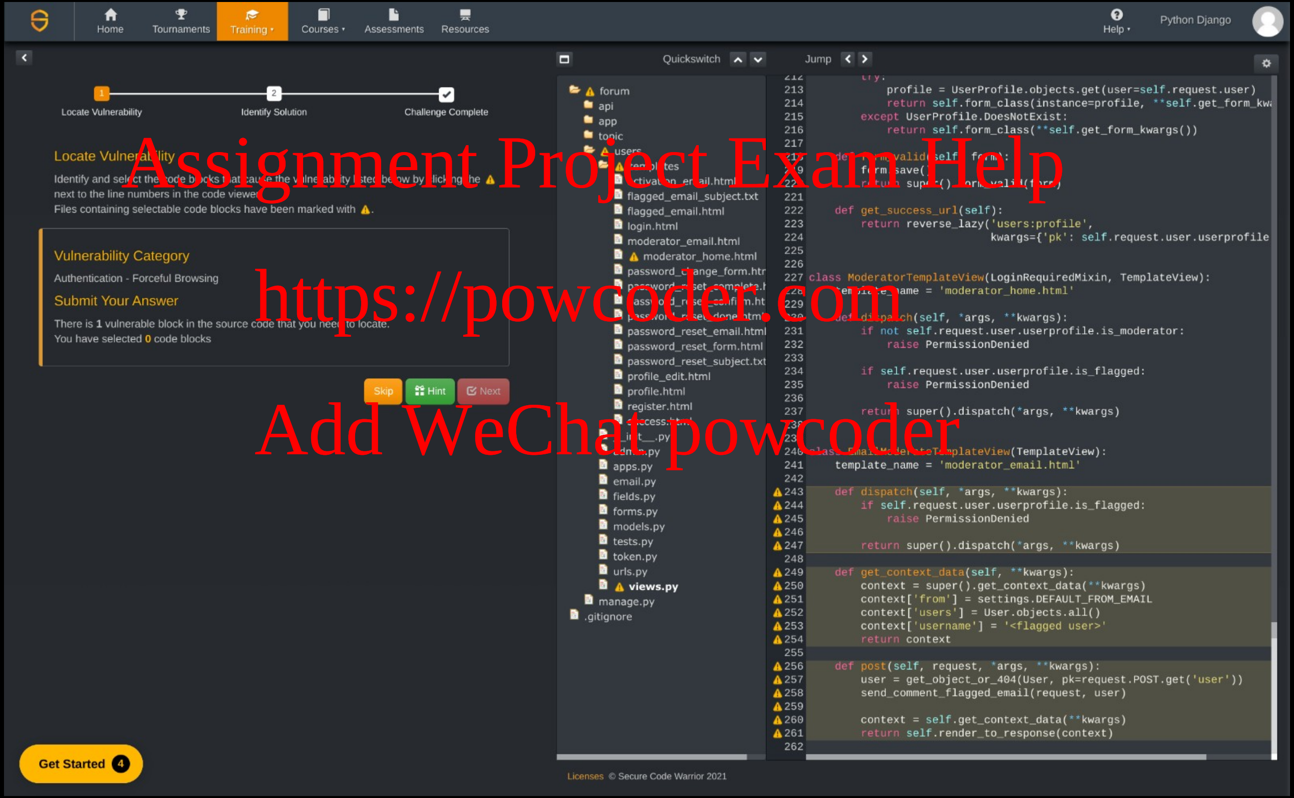
- We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- That is, while there is value in the items on the right, we value the items on the left more.

… Scrums? Kanban? Sprints? Backlog grooming?

Agile cycle

Phase	Inputs	Outcomes
Backlog	Developer training	Security prioritised
Design	Secrets management	Secure persistency
Development	Software composition analysis	Secure dependencies
Testing	Static & dynamic analysis	Fix bugs
Deployment	Containerisation, hardening	Defence in depth
Review	Root cause analysis	Bug class eradication

Developer training



1 Locate Vulnerability — 2 Identify Solution — ✓ Challenge Complete

Locate Vulnerability

Identify and select the code blocks that cause the vulnerability listed below by clicking the ⚠ next to the line numbers in the code viewer.
Files containing selectable code blocks have been marked with ⚠.

Vulnerability Category

Authentication - Forceful Browsing

Submit Your Answer

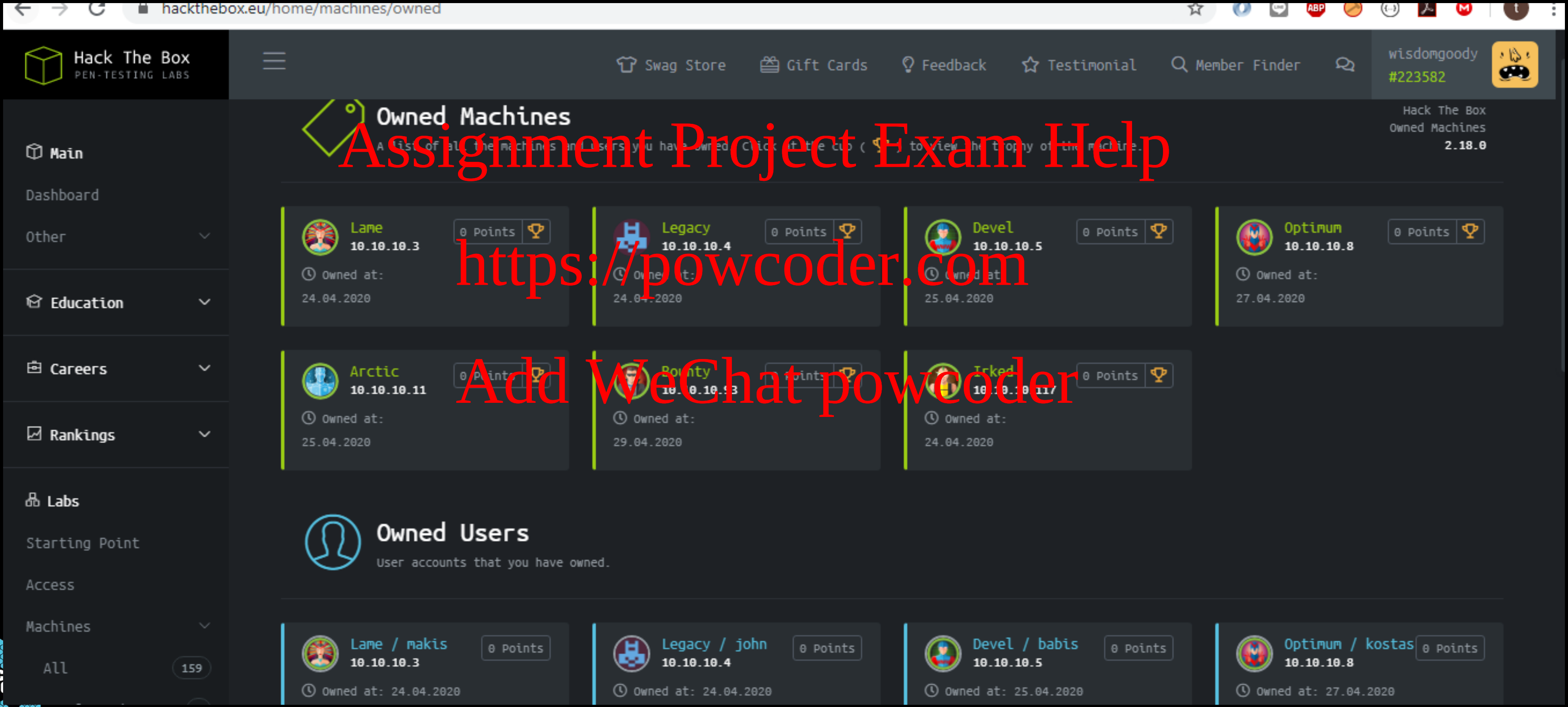
There is **1** vulnerable block in the source code that you need to locate.
You have selected **0** code blocks

```
⚠ forum
    api
    app
    topic
    ⚠ users
        ⚠ templates
            activation_email.html
            flagged_email_subject.txt
            flagged_email.html
            login.html
            moderator_email.html
            ⚠ moderator_home.html
            password_change_form.htm
            password_reset_complete.h
            password_reset_confirm.ht
            password_reset_done.html
            password_reset_email.html
            password_reset_form.html
            password_reset_subject.txt
            profile_edit.html
            profile.html
            register.html
            success.html
        __init__.py
        admin.py
        apps.py
        email.py
        fields.py
        forms.py
        models.py
        tests.py
        token.py
        urls.py
        ⚠ views.py
    manage.py
.gitignore
```

```python
        try:
            profile = UserProfile.objects.get(user=self.request.user)
            return self.form_class(instance=profile, **self.get_form_kwa
        except UserProfile.DoesNotExist:
            return self.form_class(**self.get_form_kwargs())

    def form_valid(self, form):
        form.save()
        return super().form_valid(form)

    def get_success_url(self):
        return reverse_lazy('users:profile',
                            kwargs={'pk': self.request.user.userprofile


class ModeratorTemplateView(LoginRequiredMixin, TemplateView):
    template_name = 'moderator_home.html'

    def dispatch(self, *args, **kwargs):
        if not self.request.user.userprofile.is_moderator:
            raise PermissionDenied

        if self.request.user.userprofile.is_flagged:
            raise PermissionDenied

        return super().dispatch(*args, **kwargs)


class EmailModerateTemplateView(TemplateView):
    template_name = 'moderator_email.html'

    def dispatch(self, *args, **kwargs):
        if self.request.user.userprofile.is_flagged:
            raise PermissionDenied

        return super().dispatch(*args, **kwargs)

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['from'] = settings.DEFAULT_FROM_EMAIL
        context['users'] = User.objects.all()
        context['username'] = '<flagged user>'
        return context

    def post(self, request, *args, **kwargs):
        user = get_object_or_404(User, pk=request.POST.get('user'))
        send_comment_flagged_email(request, user)

        context = self.get_context_data(**kwargs)
        return self.render_to_response(context)

```

Licenses © Secure Code Warrior 2021

Developer training



Owned Machines

A list of all the machines and users you have owned. Click at the cup (🏆) to view the trophy of the machine.

Hack The Box
Owned Machines
2.18.0

Lame 10.10.10.3 0 Points — Owned at: 24.04.2020

Legacy 10.10.10.4 0 Points — Owned at: 24.04.2020

Devel 10.10.10.5 0 Points — Owned at: 25.04.2020

Optimum 10.10.10.8 0 Points — Owned at: 27.04.2020

Arctic 10.10.10.11 0 Points — Owned at: 25.04.2020

Bounty 10.10.10.93 0 Points — Owned at: 29.04.2020

Irked 10.10.10.117 0 Points — Owned at: 24.04.2020

Owned Users

User accounts that you have owned.

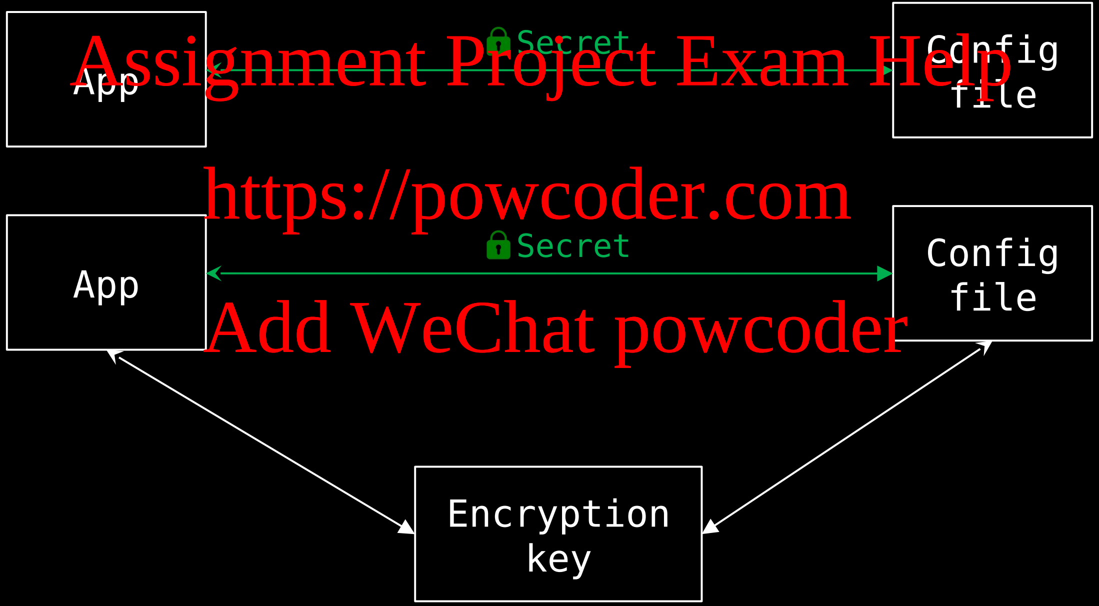
Lame / makis 10.10.10.3 0 Points — Owned at: 24.04.2020

Legacy / john 10.10.10.4 0 Points — Owned at: 24.04.2020

Devel / babis 10.10.10.5 0 Points — Owned at: 25.04.2020

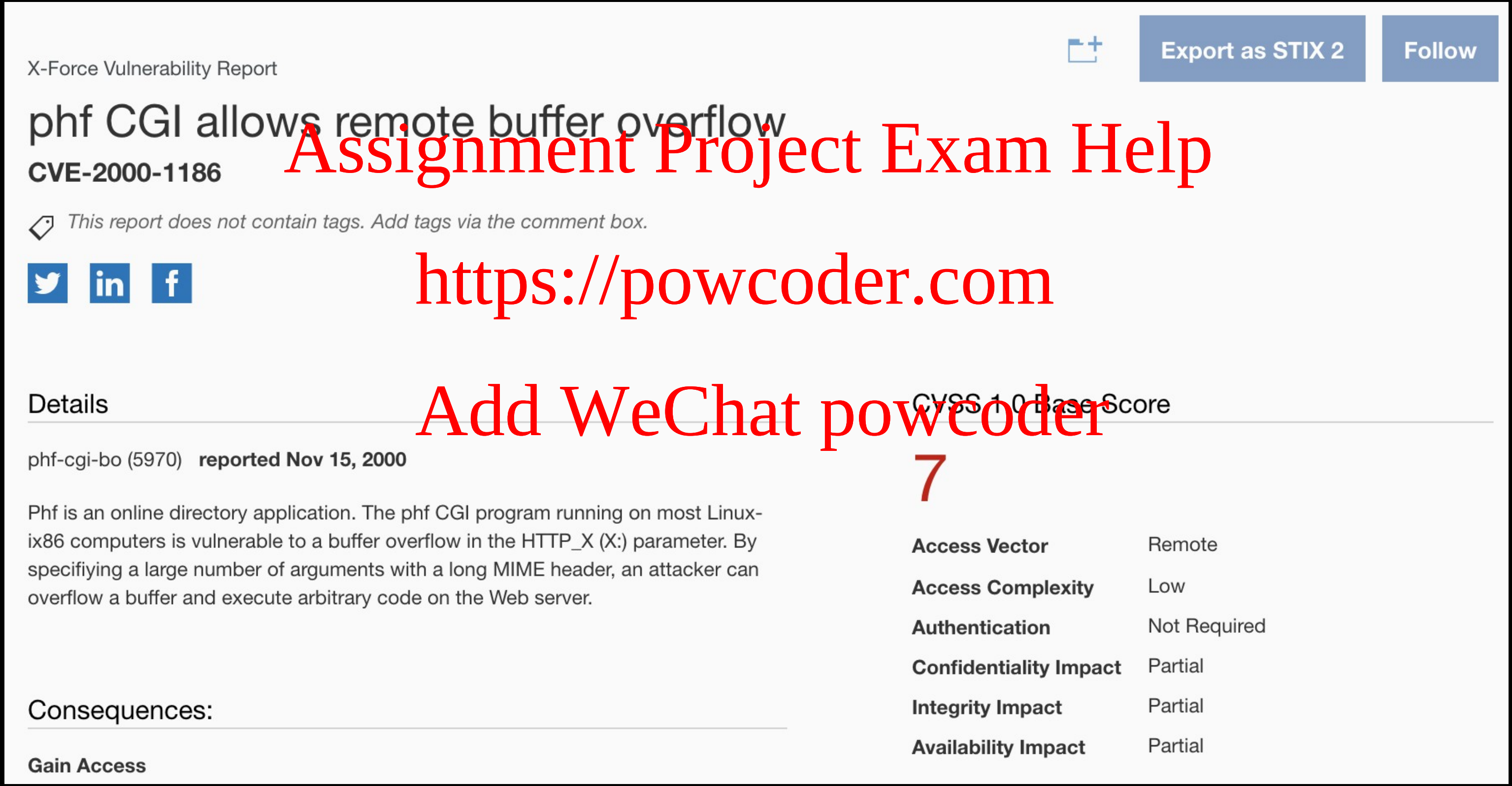
Optimum / kostas 10.10.10.8 0 Points — Owned at: 27.04.2020

Secrets management



- Password vaults are the current best solution

Common Vulnerability Enumeration CVE



X-Force Vulnerability Report

phf CGI allows remote buffer overflow

CVE-2000-1186

This report does not contain tags. Add tags via the comment box.

Details

phf-cgi-bo (5970) **reported Nov 15, 2000**

Phf is an online directory application. The phf CGI program running on most Linux-ix86 computers is vulnerable to a buffer overflow in the HTTP_X (X:) parameter. By specifiying a large number of arguments with a long MIME header, an attacker can overflow a buffer and execute arbitrary code on the Web server.

Consequences:

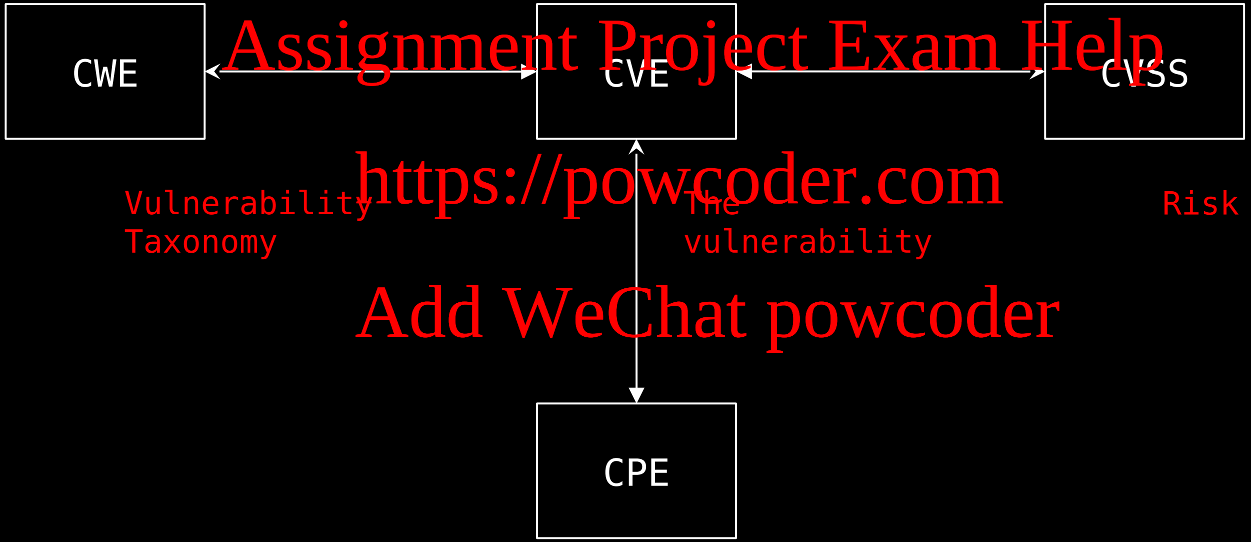
Gain Access

CVSS 1.0 Base Score

7

Access Vector	Remote
Access Complexity	Low
Authentication	Not Required
Confidentiality Impact	Partial
Integrity Impact	Partial
Availability Impact	Partial

OVAL & NIST NVD



Vulnerability Taxonomy

The vulnerability

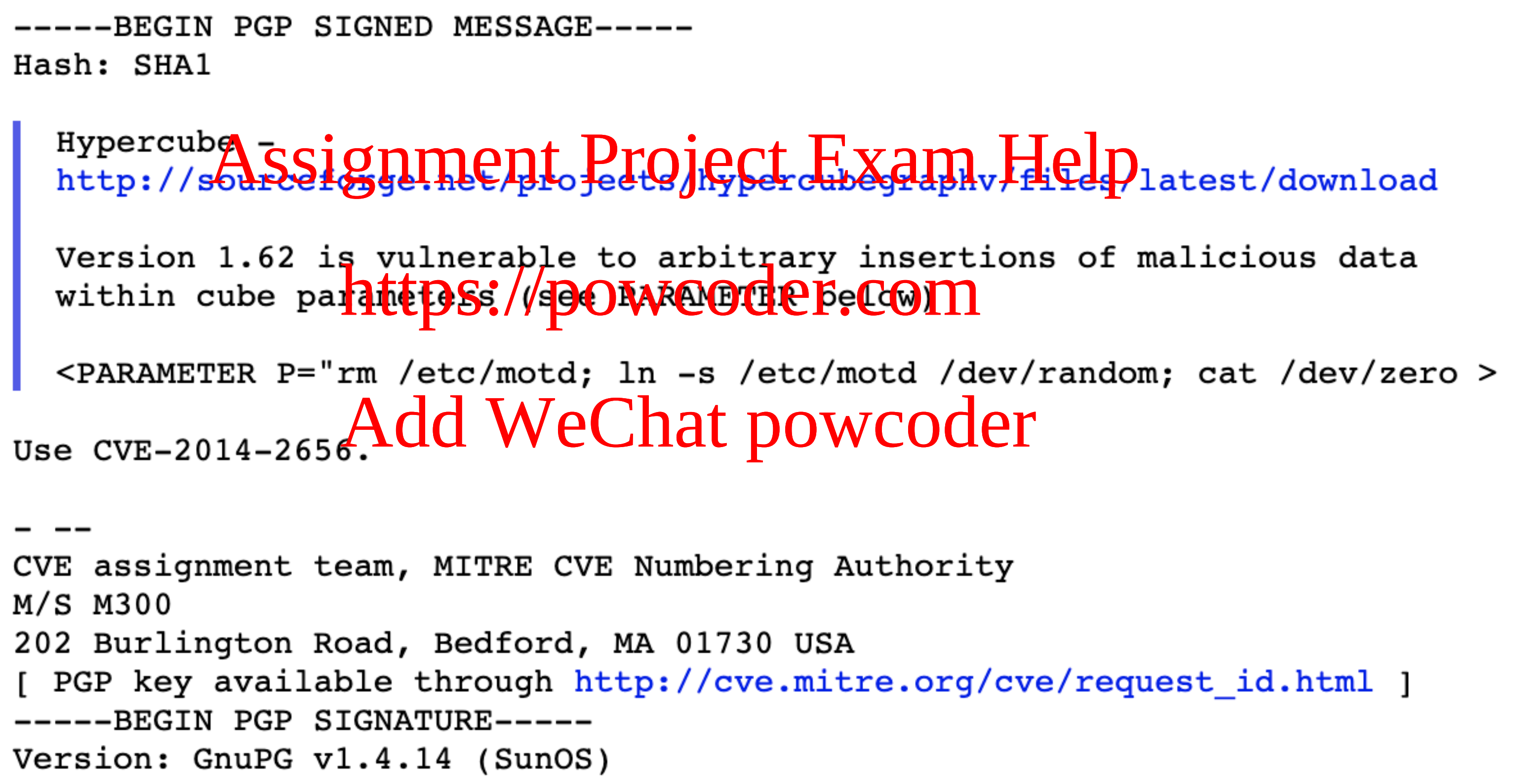
Risk rating

What is vulnerable?

NVD Example

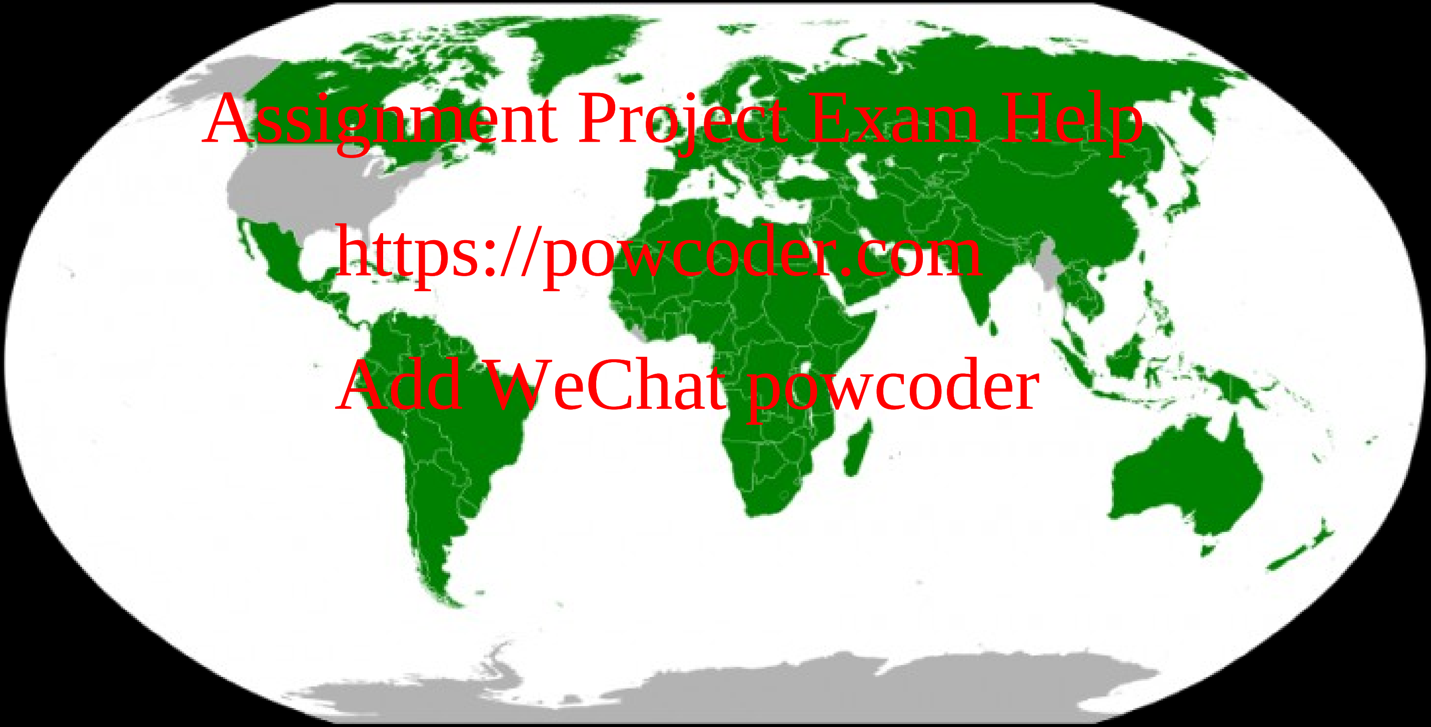
- https://nvd.nist.gov/vuln/detail/CVE-2014-0003

NVD Problems



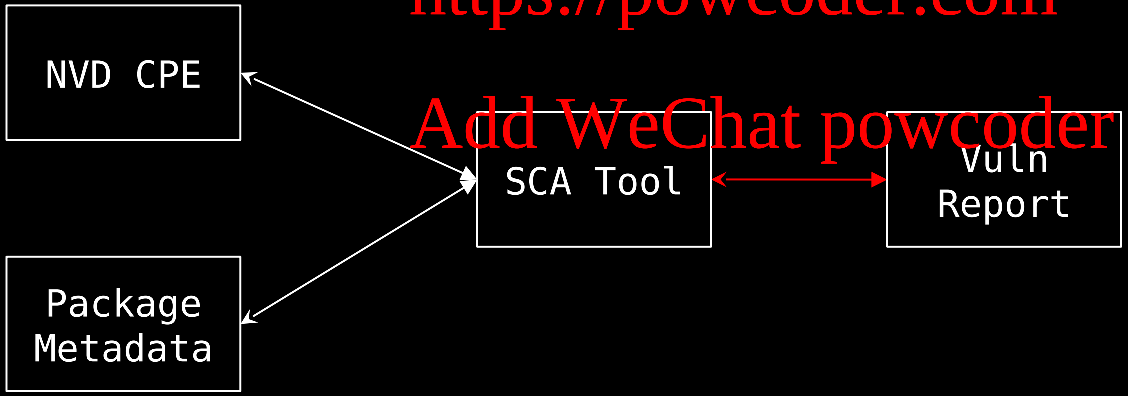
```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

  Hypercube -
  http://sourceforge.net/projects/hypercubegraphv/files/latest/download

  Version 1.62 is vulnerable to arbitrary insertions of malicious data
  within cube parameters (see PARAMETER below)

  <PARAMETER P="rm /etc/motd; ln -s /etc/motd /dev/random; cat /dev/zero >

Use CVE-2014-2656.


- --
CVE assignment team, MITRE CVE Numbering Authority
M/S M300
202 Burlington Road, Bedford, MA 01730 USA
[ PGP key available through http://cve.mitre.org/cve/request_id.html ]
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.14 (SunOS)
```

NVD Problems



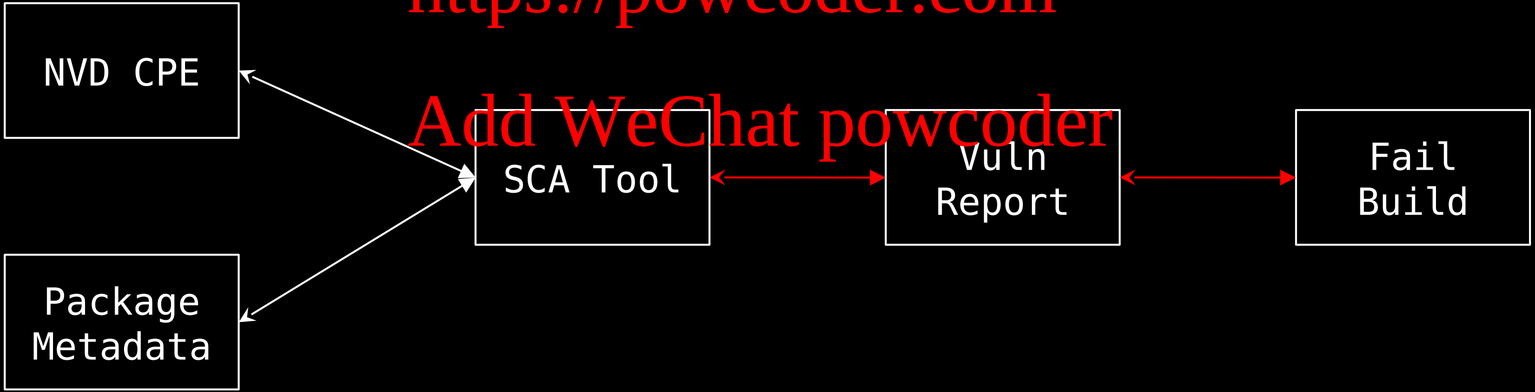
Dependency identification

- NVD CPE identifies known vulnerable versions
- Package metadata identifies version used
- SCA tool attempts to match the two and identify known vulns



Dependency identification in Java

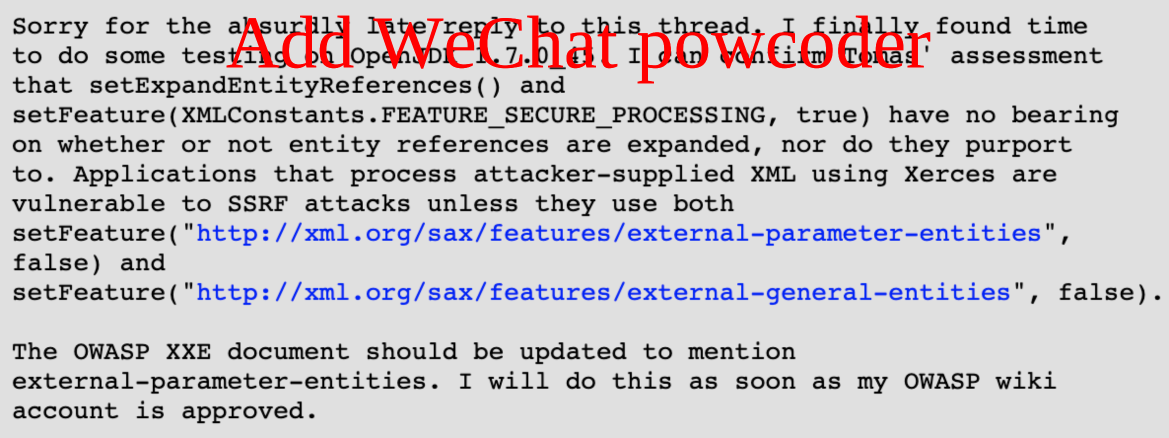
- NVD:
  - cpe:/a:springsource:spring_framework:3.2.0
    cpe:/a:pivotal:spring_framework:3.2.0
    cpe:/a:pivotal_software:spring_framework:3.2.0
- GAV:
  - org.springframework:spring-core:3.2.0.RELEASE



Source code analysis

```
$ grep -L "parameter-entities" $(grep -l -R "general-entities" *)
resteasy-jaxrs-2.3.2.Final/providers/jaxb/src/main/java/org/jboss/resteasy/plugins/providers/jaxb/ExternalEntityUnmarshaller.java
```

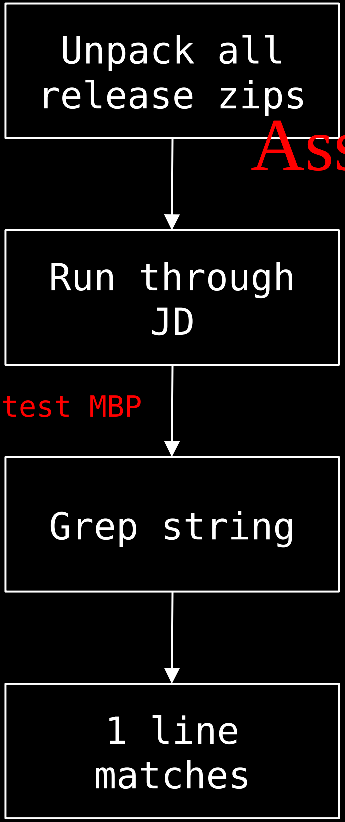
https://w 03/5



Sorry for the absurdly late reply to this thread. I finally found time to do some testing on OpenJDK 1.7.0_45. I can confirm Tomas' assessment that setExpandEntityReferences() and setFeature(XMLConstants.FEATURE_SECURE_PROCESSING, true) have no bearing on whether or not entity references are expanded, nor do they purport to. Applications that process attacker-supplied XML using Xerces are vulnerable to SSRF attacks unless they use both setFeature("http://xml.org/sax/features/external-parameter-entities", false) and setFeature("http://xml.org/sax/features/external-general-entities", false).

The OWASP XXE document should be updated to mention external-parameter-entities. I will do this as soon as my OWASP wiki account is approved.

Source code analysis



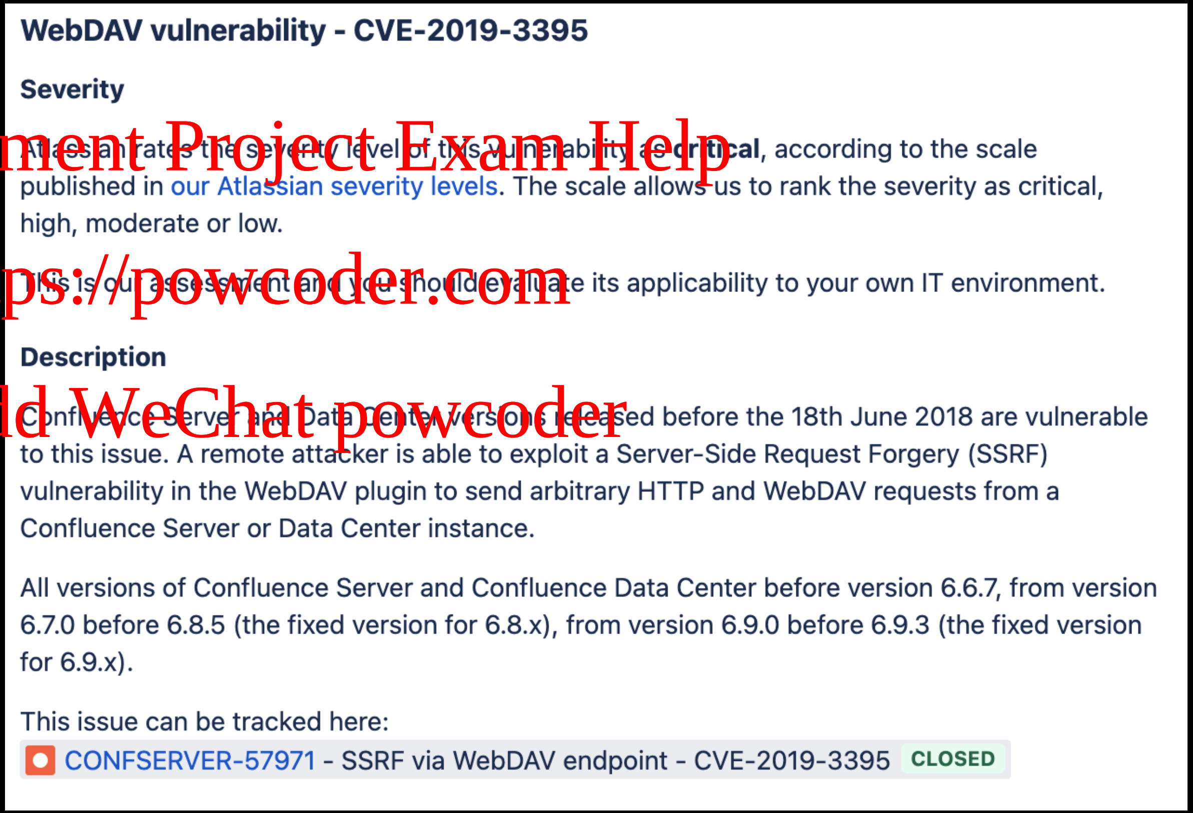
Unpack all release zips

Run through JD

~3 hrs on latest MBP

Grep string

1 line matches



WebDAV vulnerability - CVE-2019-3395

Severity

Atlassian rates the severity level of this vulnerability as **critical**, according to the scale published in our Atlassian severity levels. The scale allows us to rank the severity as critical, high, moderate or low.

This is our assessment and you should evaluate its applicability to your own IT environment.

Description

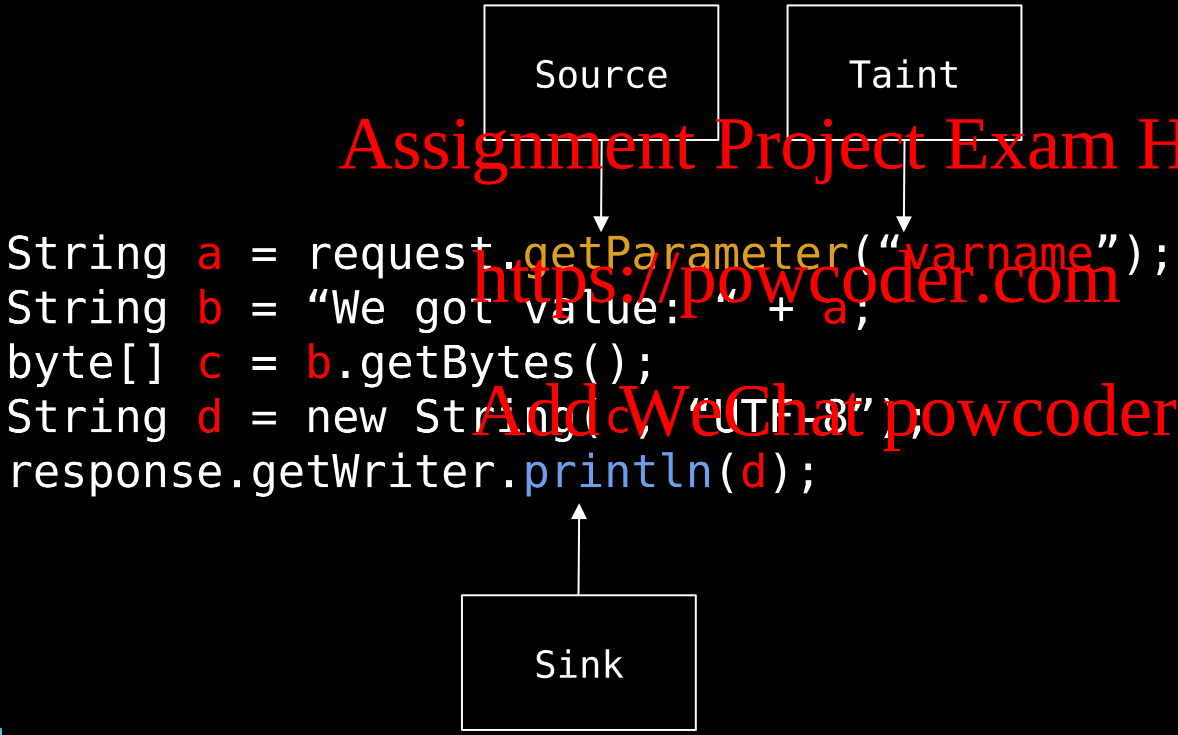
Confluence Server and Data Center versions released before the 18th June 2018 are vulnerable to this issue. A remote attacker is able to exploit a Server-Side Request Forgery (SSRF) vulnerability in the WebDAV plugin to send arbitrary HTTP and WebDAV requests from a Confluence Server or Data Center instance.

All versions of Confluence Server and Confluence Data Center before version 6.6.7, from version 6.7.0 before 6.8.5 (the fixed version for 6.8.x), from version 6.9.0 before 6.9.3 (the fixed version for 6.9.x).

This issue can be tracked here:

CONFSERVER-57971 - SSRF via WebDAV endpoint - CVE-2019-3395 CLOSED

Sources, sinks & taints



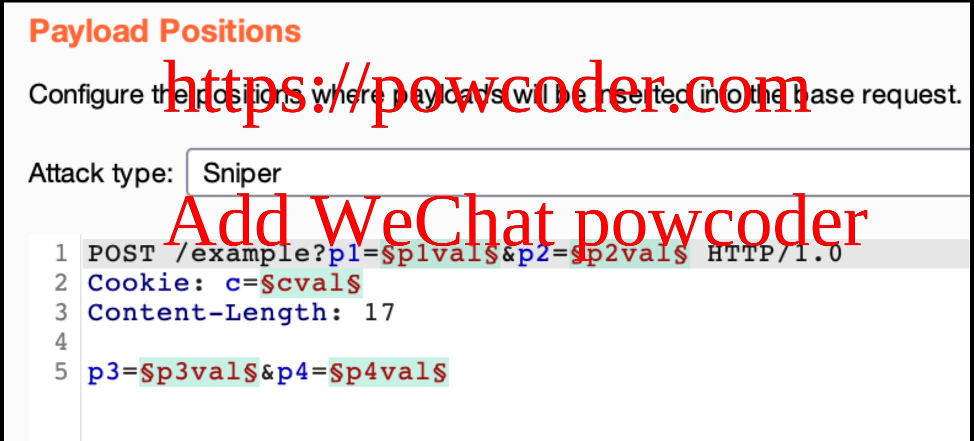
```
String a = request.getParameter(“varname”);
String b = “We got value: ” + a;
byte[] c = b.getBytes();
String d = new String(c, “UTF-8”);
response.getWriter.println(d);
```

Static application security testing

Pros	Cons
Find & fix vulns early	Massive false positives
Identify vulns in configuration & conditions	Manual triage & exploitation
Open source tools available	Commercial deployments = $ $$
Potential for bug class eradication	Complexity of tweaking rules

Dynamic application security testing

AKA DAST. Many tools, big commercial ones include Netsparker, Tenable, CheckMarx and Veracode.



Payload Positions

Configure the positions where payloads will be inserted into the base request.

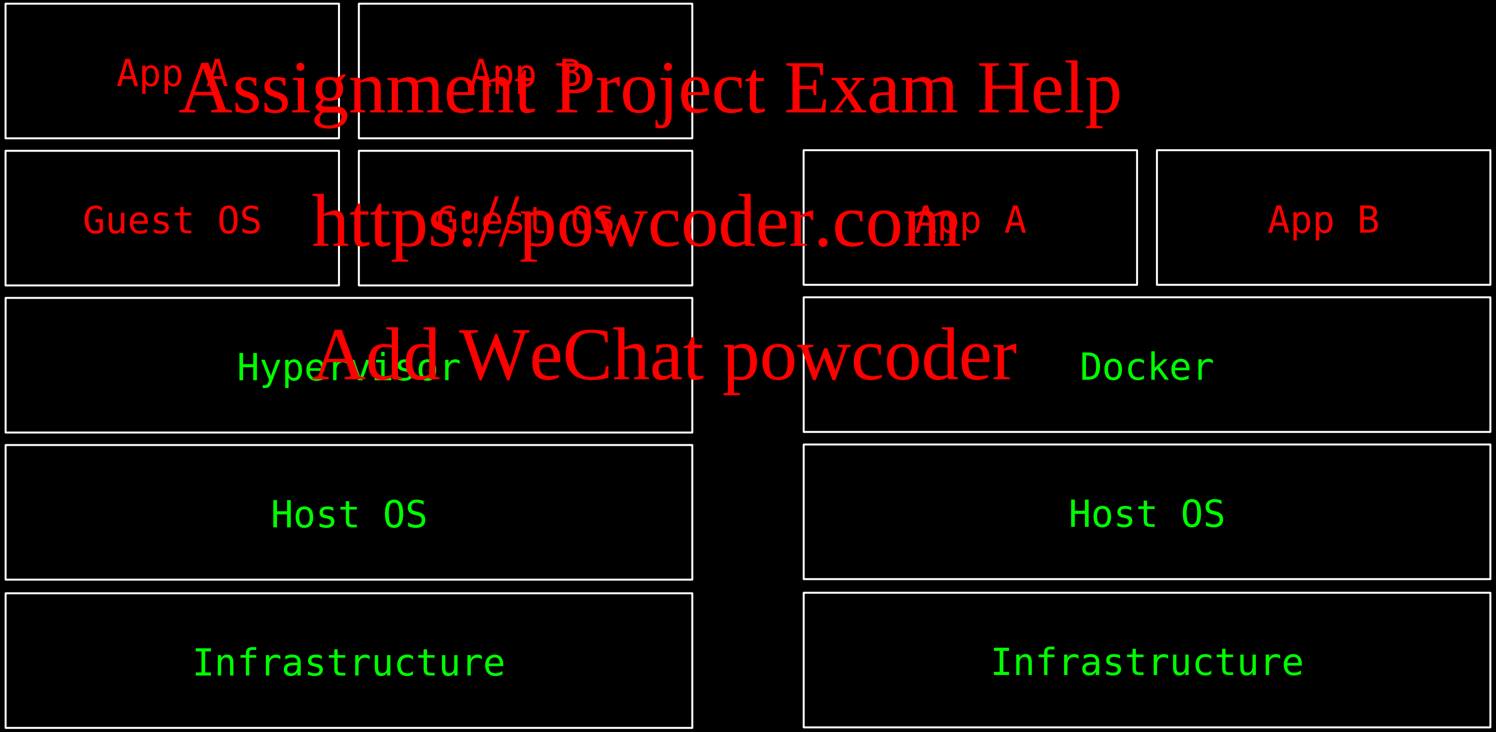
Attack type: Sniper

```
POST /example?p1=§p1val§&p2=§p2val§ HTTP/1.0
Cookie: c=§cval§
Content-Length: 17

p3=§p3val§&p4=§p4val§
```

Dynamic application security testing

Pros	Cons
Scanning of live targets	Data corruption
Language independent	Cannot read config files
Cloud based deployment	Cannot understand complex dynamic client/server
Less false positives than SAST	Relies on configuration to map attack surface

Virtualisation vs containerisation



Container breakout CVE-2019-5736

RunC is a container runtime originally developed as part of Docker and later extracted out as a separate open source tool and library. As a "low level" container runtime, runC is mainly used by "high level" container runtimes (e.g. Docker) to spawn and run containers, although it can be used as a stand-alone tool. "High level" container runtimes like Docker will normally implement functionalities such as image creation and management and will use runC to handle tasks related to running containers – creating a container, attaching a process to an existing container (docker exec) and so on.

Credit:
https://unit42.paloaltonetworks.com/breaking-docker-via-runc-explaining-cve-2019-5736/

Container breakout CVE-2019-5736

procfs is a virtual fs in Linux that presents information about processes, mounted to /proc. It can be thought of as an interface to system data that the kernel exposes as a filesystem. Each process has its own directory in procfs, at /proc/[pid]

/proc/self points to the current process. Each process’s directory contains information on the process. For the vulnerability, the relevant items are:

- /proc/self/exe – a symbolic link to the executable file the process is running
- /proc/self/fd – a directory containing the file descriptors open by the process.

For example, using ls /proc/self one can see that /proc/self/exe points to the ‘ls’ executable.

Container breakout CVE-2019-5736

procfs is a virtual fs in Linux that presents information about processes, mounted to /proc. It can be thought of as an interface to system data that the kernel exposes as a filesystem. Each process has its own directory in procfs, at /proc/[pid]

/proc/self points to the current process. Each process’s directory contains information on the process. For the vulnerability, the relevant items are:

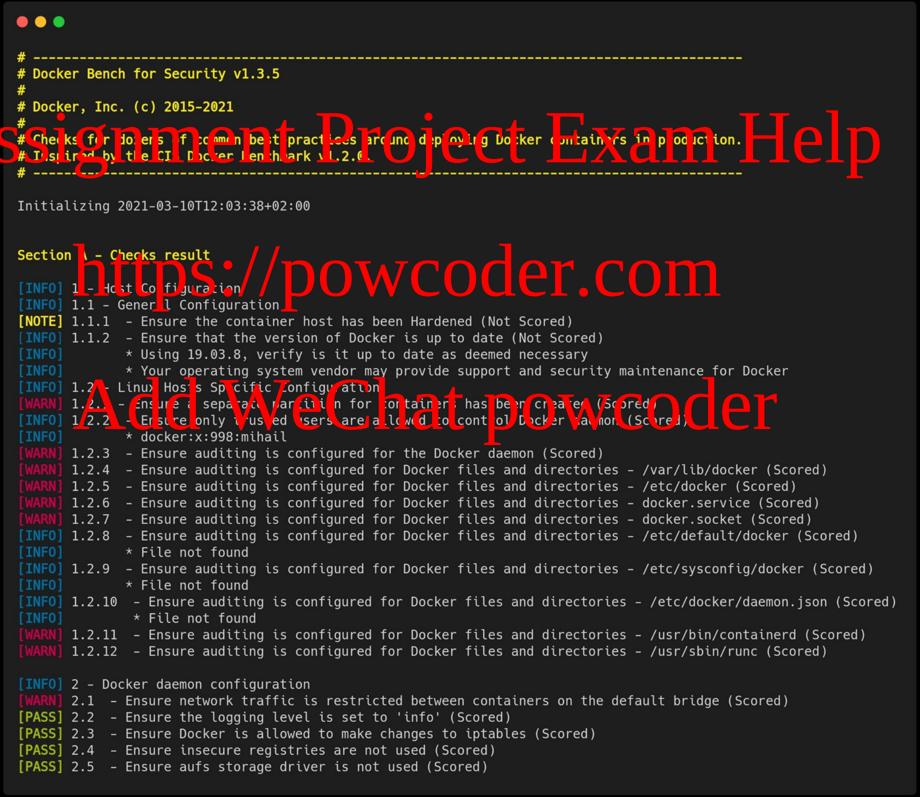
- /proc/self/exe – a symbolic link to the executable file the process is running
- /proc/self/fd – a directory containing the file descriptors open by the process.

For example, using ls /proc/self one can see that /proc/self/exe points to the ‘ls’ executable.

Container breakout CVE-2019-5736

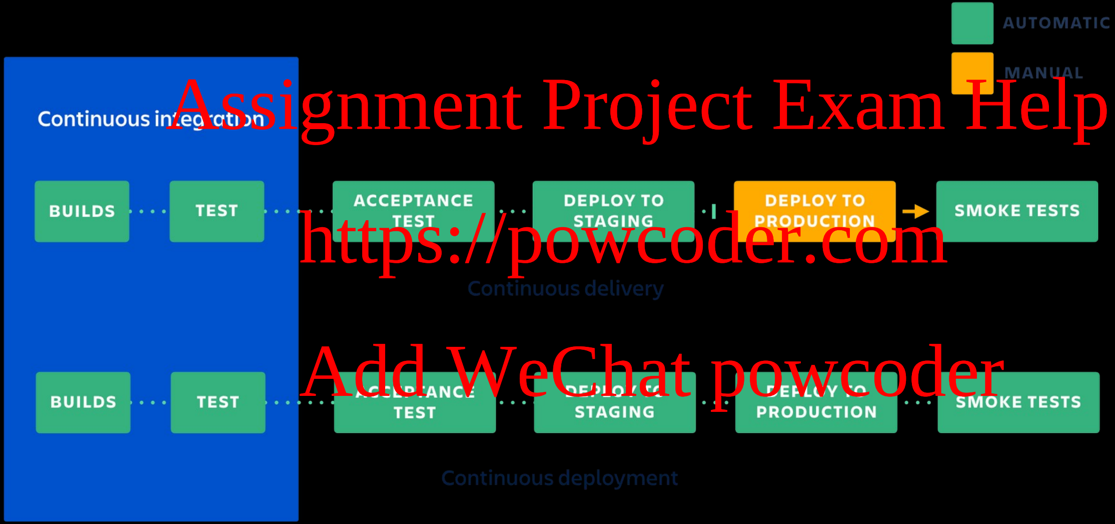
- Anattacker can trick runC into executing itself by asking it to run /proc/self/exe, which is a symbolic link to the runC binary on the host.
- An attacker with root access in the container can then use /proc/[runc-pid]/exe as a reference to the runC binary on the host and overwrite it.
- Root access in the container is required to perform this attack as the runC binary is owned by root.
- The next time runC is executed, the attacker will achieve code execution on the host.
- Since runC is normally run as root (e.g. by the Docker daemon), the attacker will gain root access on the host.

docker-bench-security



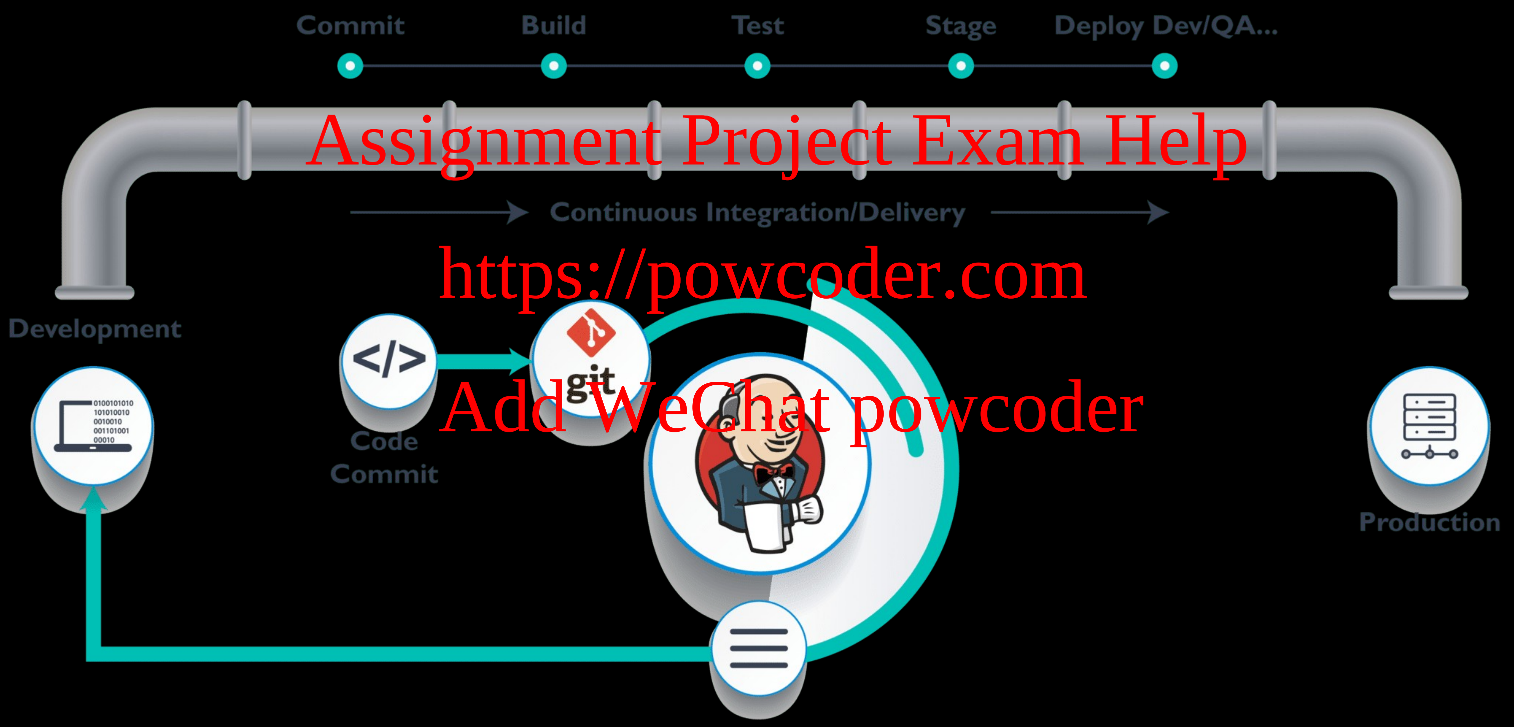
```
# --------------------------------------------------------------------------------------------
# Docker Bench for Security v1.3.5
#
# Docker, Inc. (c) 2015-2021
#
# Checks for dozens of common best-practices around deploying Docker containers in production.
# Inspired by the CIS Docker Benchmark v1.2.0.
# --------------------------------------------------------------------------------------------

Initializing 2021-03-10T12:03:38+02:00


Section A - Checks result

[INFO] 1 - Host Configuration
[INFO] 1.1 - General Configuration
[NOTE] 1.1.1  - Ensure the container host has been Hardened (Not Scored)
[INFO] 1.1.2  - Ensure that the version of Docker is up to date (Not Scored)
[INFO]        * Using 19.03.8, verify is it up to date as deemed necessary
[INFO]        * Your operating system vendor may provide support and security maintenance for Docker
[INFO] 1.2 - Linux Hosts Specific Configuration
[WARN] 1.2.1 - Ensure a separate partition for containers has been created (Scored)
[INFO] 1.2.2 - Ensure only trusted users are allowed to control Docker daemon (Scored)
[INFO]        * docker:x:998:mihail
[WARN] 1.2.3  - Ensure auditing is configured for the Docker daemon (Scored)
[WARN] 1.2.4  - Ensure auditing is configured for Docker files and directories - /var/lib/docker (Scored)
[WARN] 1.2.5  - Ensure auditing is configured for Docker files and directories - /etc/docker (Scored)
[WARN] 1.2.6  - Ensure auditing is configured for Docker files and directories - docker.service (Scored)
[WARN] 1.2.7  - Ensure auditing is configured for Docker files and directories - docker.socket (Scored)
[INFO] 1.2.8  - Ensure auditing is configured for Docker files and directories - /etc/default/docker (Scored)
[INFO]        * File not found
[INFO] 1.2.9  - Ensure auditing is configured for Docker files and directories - /etc/sysconfig/docker (Scored)
[INFO]        * File not found
[INFO] 1.2.10  - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json (Scored)
[INFO]         * File not found
[WARN] 1.2.11  - Ensure auditing is configured for Docker files and directories - /usr/bin/containerd (Scored)
[WARN] 1.2.12  - Ensure auditing is configured for Docker files and directories - /usr/sbin/runc (Scored)

[INFO] 2 - Docker daemon configuration
[WARN] 2.1  - Ensure network traffic is restricted between containers on the default bridge (Scored)
[PASS] 2.2  - Ensure the logging level is set to 'info' (Scored)
[PASS] 2.3  - Ensure Docker is allowed to make changes to iptables (Scored)
[PASS] 2.4  - Ensure insecure registries are not used (Scored)
[PASS] 2.5  - Ensure aufs storage driver is not used (Scored)
```

Continuous integration|deployment



Source: atlassian.com

Continuous integration|deployment



READING MATERIAL (REFERENCE)

Find-sec-bugs
https://find-sec-bugs.github.io/

Tracking vulnerable JARs
https://www.slideshare.net/davidjorm/tracking-vulnerable-jars

OWASP dependency check
https://owasp.org/www-project-dependency-check/

OWASP ZAP
https://owasp.org/www-project-zap/

Jenkins
https://www.jenkins.io/

Docker-bench-security
https://github.com/docker/docker-bench-security

WEEK 9 ASSESSMENT

- Exam question based on provided scenario
- Similar in structure to a job interview question
- Answer will be a few paragraphs of text

Please call out if you get stuck.
Support one another, your tutors are here to help!

THANKS FOR LISTENING TO US RANT!

questions? slack / email / come talk to us