

# N S 2022

ACCELERATING INNOVATION

# Security Best Practices

*Bill Smith*

*Lead Software Architect - Tridium*



**NS2022**  
ACCELERATING INNOVATION



# NS2022

ACCELERATING INNOVATION

## Overview

- Tools of the Trade
  - Threat Modeling
  - Code Coverage/Unit Tests
  - Code Reviews
  - Static Code Analysis
  - Third Party Components
  - Penetration Testing
- Assessing Risk
  - CVSS
  - Beyond CVSS
- Lessons Learned

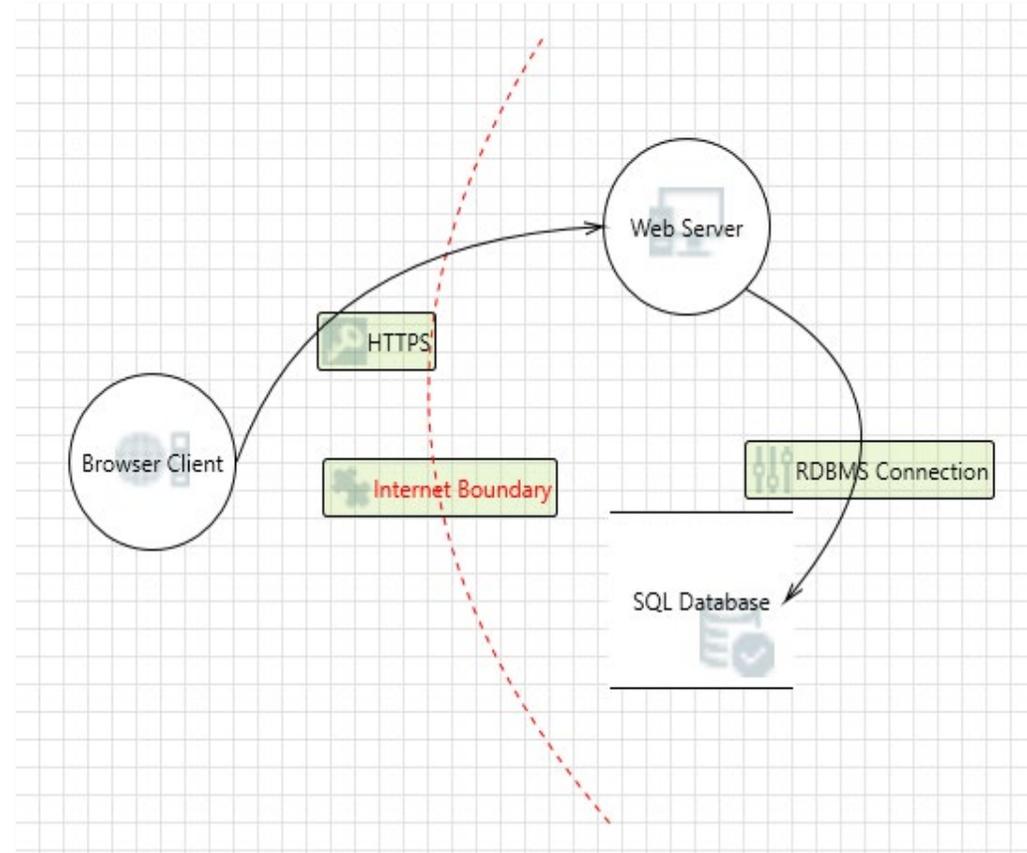
# Tools of the Trade

- Threat Modeling
- Code Coverage/Unit Tests
- Code Reviews
- Static Code Analysis
- Third Party Components
- Pen Testing



# Threat Modeling

- MS Threat Modeling Tool is the tool of choice for us
- Threat model diagram derived from network diagram
- Auto generates risks based on interactions
- Risks outside the automatic ones can be added



# Threat Modeling

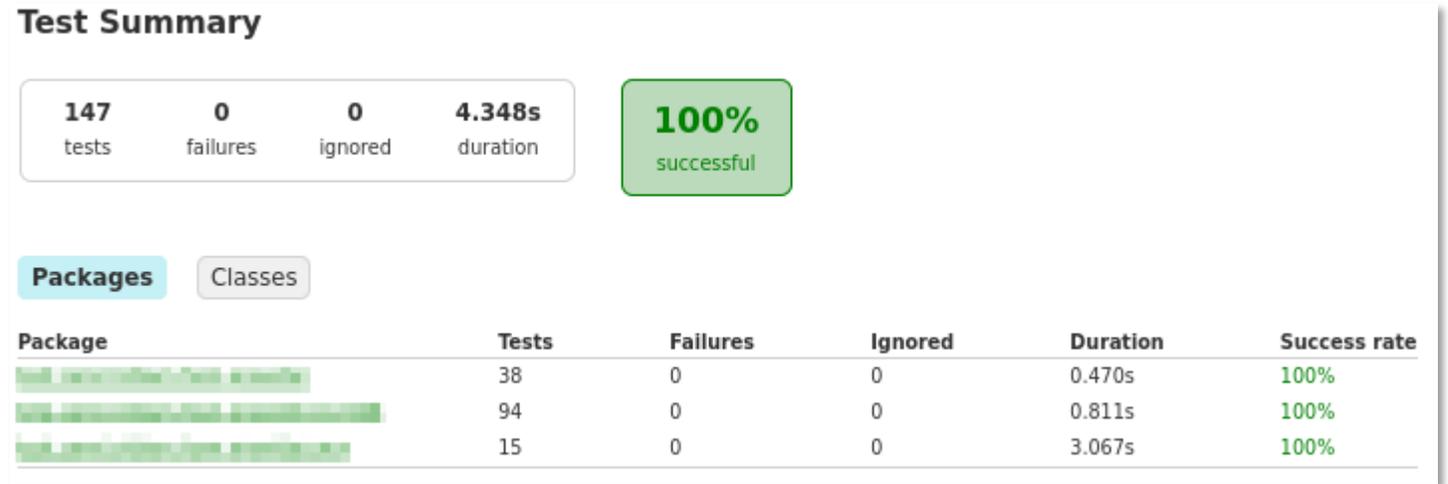
- Threat List shows list of identified threats.
- Threat properties provide status and justification of identified threat.

| Threat List |                  |                      |                       |                  |  |
|-------------|------------------|----------------------|-----------------------|------------------|--|
| ID          | Diagram          | Changed By           | Last Modified         | State            | Title                                    |
| 0           | Diagram 1        | GLOBAL\E33397        | 3/29/2022 10:17:4     | Mitigated        | Spoofing the Browser Client Process      |
| <b>1</b>    | <b>Diagram 1</b> | <b>GLOBAL\E33397</b> | <b>3/29/2022 10:1</b> | <b>Mitigated</b> | <b>Cross Site Scripting</b>              |
| 2           | Diagram 1        | GLOBAL\E33397        | 3/29/2022 10:18:0     | Mitigated        | Potential Data Repudiation by Web Server |

| Threat Properties |   |
|-------------------|---|
| ID: 1             | Diagram: Diagram 1  |
| Status:           | Mitigated   |
| Title:            | Cross Site Scripting  |
| Category:         | Tampering   |
| Description:      | The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input. |
| Justification:    | All input is santized by application.   |
| Interaction:      | HTTPS   |
| Priority:         | High  |

# Code Coverage/Unit Tests

- Unit Test Tools
  - JUnit
  - TestNG
  - Jasmine/Karma
- Code Coverage Tools
  - JaCoCo
  - SonarQube
  - Istanbul



### niagara-test-provider

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cx |
|---------|---------------------|------|-----------------|------|--------|----|
| ...     |                     | 91%  |                 | 42%  | 59     | 11 |
| ...     |                     | 77%  |                 | 66%  | 16     | 4  |
| ...     |                     | 84%  |                 | 75%  | 14     | 1  |
| Total   | 790 of 7,039        | 88%  | 100 of 224      | 55%  | 89     | 23 |

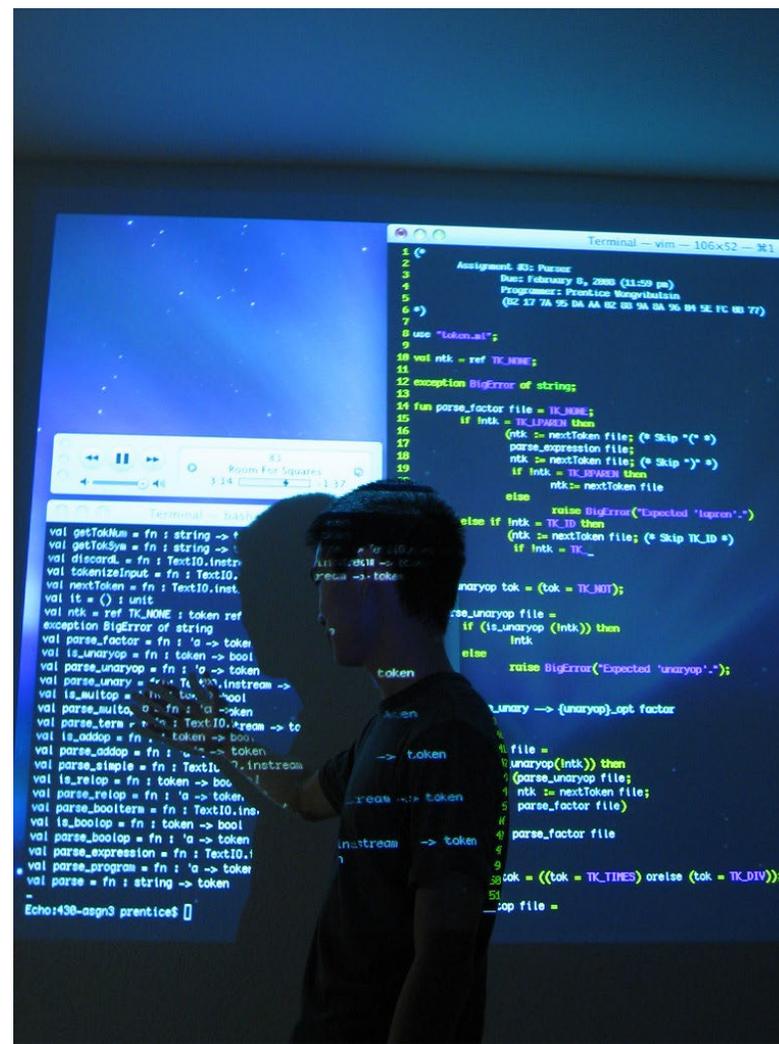
# Code Reviews

Some things to consider:

- Coding Standards
- Logging of Sensitive Data
- Input Validation
- Returning Privileged Information
- Hardcoded Credentials

Some Niagara specific things:

- Permissions Model
- Security Audit Log
- Security Facet
- Security Dashboard Card



# Static Code Analysis

- Tools
  - Coverity
  - SonarQube
- Functions
  - Security
  - Quality

The screenshot displays a static code analysis tool interface. At the top, it shows 'Issues: By Snapshot' and 'Filters: Streams'. Below this is a table of issues with columns for CID, Type, Impact, Status, First Detected, Component, Action, Category, and File. The table lists several issues, with CID 64252 highlighted. Below the table, it indicates '1 of 487 issues selected'. The main area shows a code snippet with a constructor function. A red box highlights a specific issue: 'CID 64252 (#1 of 6): Uninitialized pointer field (UNINIT\_CTOR)'. The description states: '2. uninitialized: Non-static class member `_M_single_bucket` is not initialized in this constructor nor in any functions that it calls.'

| CID   | Type                          | Impact | Status | First Detected | Component | Action    | Category              | File   |
|-------|-------------------------------|--------|--------|----------------|-----------|-----------|-----------------------|--------|
| 64255 | Uninitialized pointer field   | Medium | New    | 02/17/16       | [Link]    | Undecided | Uninitialized members | [Link] |
| 64254 | Recursion in included headers | Low    | New    | 02/17/16       | [Link]    | Undecided | Build system issues   | [Link] |
| 64253 | Uninitialized scalar field    | Medium | New    | 02/17/16       | [Link]    | Undecided | Uninitialized members | [Link] |
| 64252 | Uninitialized pointer field   | Medium | New    | 02/17/16       | [Link]    | Ignore    | Uninitialized members | [Link] |
| 64251 | Resource leak                 | High   | New    | 02/17/16       | [Link]    | Undecided | Resource leaks        | [Link] |
| 64250 | Recursion in included headers | Low    | New    | 02/17/16       | [Link]    | Undecided | Build system issues   | [Link] |
| 64249 | Uncaught exception            | Medium | New    | 02/17/16       | [Link]    | Undecided | Error handling issues | [Link] |
| 64248 | Uninitialized pointer field   | Medium | New    | 02/17/16       | [Link]    | Undecided | Uninitialized members | [Link] |
| 64247 | Uncaught exception            | Medium | New    | 02/17/16       | [Link]    | Undecided | Error handling issues | [Link] |

```
793     const allocator_type& __a)
794 :   _hashtable_base(__exk, __h1, __h2, __h, __eq),
795   _map_base(),
796   _rehash_base(),
797   _hashtable_alloc(__node_alloc_type(__a)),
798   _M_element_count(0),
799   _M_rehash_policy()
800 {
801     _M_bucket_count = _M_rehash_policy._M_next_bkt(__bucket_hint);
802     _M_buckets = _M_allocate_buckets(_M_bucket_count);
803 }
804
805 template<typename _Key, typename _Value,
```

# Third Party Components

- Tools
  - Black Duck Hub
  - OWASP Dependency Checker
- Other Resources
  - <https://www.cvedetails.com/>
  - <https://cve.mitre.org/cve/>
  - <https://nvd.nist.gov/>

## CVE Details

The ultimate security vulnerability datasource

[Log In](#) [Register](#)

[Switch to https://](#)

[Home](#)

**Browse :**

[Vendors](#)

[Products](#)

[Vulnerabilities By Date](#)

[Vulnerabilities By Type](#)

**Reports :**

[CVSS Score Report](#)

[CVSS Score Distribution](#)

**Search :**

[Vendor Search](#)

[Product Search](#)

[Version Search](#)

[Vulnerability Search](#)

[By Microsoft References](#)

**Top 50 :**

[Vendors](#)

[Vendor Cvss Sc](#)

[Products](#)

[Product Cvss Sc](#)

[Versions](#)

**Other :**

[Microsoft Bullet](#)

[Bugtraq Entries](#)

Enter a CVE id, product, vendor, vulnerability type...

### Current CVSS Score Distribution For All Vulnerabilities

#### Distribution of all vulnerabilities by CVSS Scores

| CVSS Score | Number Of Vulnerabilities | Percentage |
|------------|---------------------------|------------|
| 0-1        | 677                       | 0.40       |
| 1-2        | 1166                      | 0.70       |
| 2-3        | 7956                      | 4.60       |
| 3-4        | 8687                      | 5.00       |
| 4-5        | 41027                     | 23.70      |

#### Vulnerability Distribution By CVSS Scores



# NIST

Information Technology Laboratory

## NATIONAL VULNERABILITY DATABASE

# Third Party Components

- Tools
  - Black Duck Hub
  - OWASP Depend
- Other Resources
  - <https://>
  - <https://cve>
  - <https://>

Validate Your Bill  
of Materials  
First!!!



# Penetration Testing

- Tools
  - BurpSuite
  - Metasploit
  - FuzzDB
  - Nmap
  - Qualys

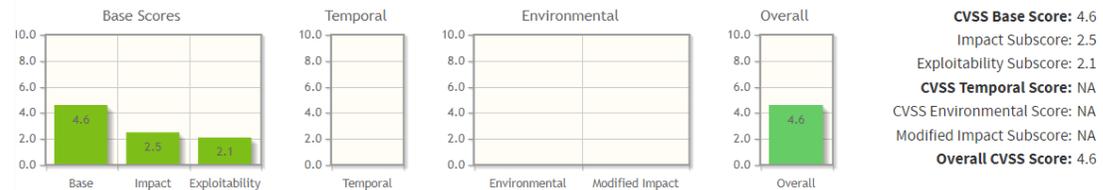


# Assessing Risk

- Primary method is CVSS (Common Vulnerability Scoring System)
- NIST Calculator (<https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>)
- Environmental scoring can help, but...

## Common Vulnerability Scoring System Calculator

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



Show Equations

CVSS v3.1 Vector

AV:A/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N

### Base Score Metrics

#### Exploitability Metrics

Attack Vector (AV)\*

Network (AV:N) **Adjacent Network (AV:A)** Local (AV:L) Physical (AV:P)

Attack Complexity (AC)\*

Low (AC:L) High (AC:H)

Privileges Required (PR)\*

None (PR:N) **Low (PR:L)** High (PR:H)

User Interaction (UI)\*

**None (UI:N)** Required (UI:R)

Scope (S)\*

**Unchanged (S:U)** Changed (S:C)

Impact Metrics

Confidentiality Impact (C)\*

None (C:N) **Low (C:L)** High (C:H)

Integrity Impact (I)\*

None (I:N) **Low (I:L)** High (I:H)

Availability Impact (A)\*

**None (A:N)** Low (A:L) High (A:H)

# Contrived Example



# Contrived Example



# Beyond CVSS

- Consider other factors in addition to CVSS, examples:
  - Fire/Life Safety
  - Alternative mitigations
  - Number of customers impacted
  - Risk of destabilizing product
  - Cost to fix



# Lessons Learned

- Relationship with engineering teams should be symbiotic, NOT adversarial
- BREAK THE BUILD can be very counter productive
- Understand WHY something is being done, don't just be a checkbox
- OWN and EMBRACE findings and tackle them with pride

# Questions?

*bsmith@tridium.com*

*<https://www.linkedin.com/in/bill-smith-vmi91/>*



**NS2022**  
ACCELERATING INNOVATION

# References/Citations

- *"Tools"* by zzpza is marked with CC BY 2.0. To view the terms, visit <https://creativecommons.org/licenses/by/2.0/?ref=openverse>
- *MS Threat Modeling Tool*, visit <https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool>
- *TestNG*, visit <https://testng.org/doc/>
- *"Code on the Wall"* by Nat W is marked with CC BY-SA 2.0. To view the terms, visit <https://creativecommons.org/licenses/by-sa/2.0/?ref=openverse>
- *BurpeSuite*, visit <https://portswigger.net/burp>
- *Metasploit*, visit <https://www.metasploit.com/>
- *Fuzzdb*, visit <https://github.com/fuzzdb-project/fuzzdb>
- *NMAP*, visit <https://nmap.org/>
- *CVE Details*, visit <https://www.cvedetails.com/>
- *CVE Mitre List*, visit <https://cve.mitre.org/cve/>
- *"Aaaagh!"* by Finizio is marked with CC BY-ND 2.0. To view the terms, visit <https://creativecommons.org/licenses/by-nd/2.0/?ref=openverse>
- *NIST Calculator*, visit <https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>
- *Water Fountain 3"* by joshme17 is marked with CC BY 2.0. To view the terms, visit <https://creativecommons.org/licenses/by/2.0/?ref=openverse>
- *"Drip."* by Instant Vantage is marked with CC BY-SA 2.0. To view the terms, visit <https://creativecommons.org/licenses/by-sa/2.0/?ref=openverse>
- *"File:Johnny-automatic-scales-of-justice.svg"* by johnny\_automatic is marked with CC0 1.0. To view the terms, visit <https://creativecommons.org/publicdomain/zero/1.0/deed.en?ref=openverse>



**NS2022**  
ACCELERATING INNOVATION