GOING WHERE NO WAFS HAVE GONE BEFORE

Andy Prow
Aura Information Security

Sam Pickles
Senior Systems Engineer, F5 Networks NZ

Agenda:

- WTF is a WAF?
- View from the Trenches
- Example Attacks and Mitigation Methods

WTF is a WAF?

Surely not another security technology?

- We already have:
  - Intrusion Prevention,
  - Firewalls,
  - Strong Authentication,
  - Patch Management
  - Vulnerability Scanning
  - VPN
  - Antivirus
  - DDoS mitigators
  - ...

Virtually every organisation has vulnerabilities

"8 out of 10 websites vulnerable to attack"

- WhiteHat "security report "

"97% of websites at immediate risk of being hacked due to vulnerabilities! 69% of vulnerabilities are client side-attacks"

- Web Application Security Consortium

"75 percent of hacks happen at the application."

- Gartner "Security at the Application Level"

"64 percent of developers are not confident in their ability to write secure applications."

- *Microsoft Developer Research*

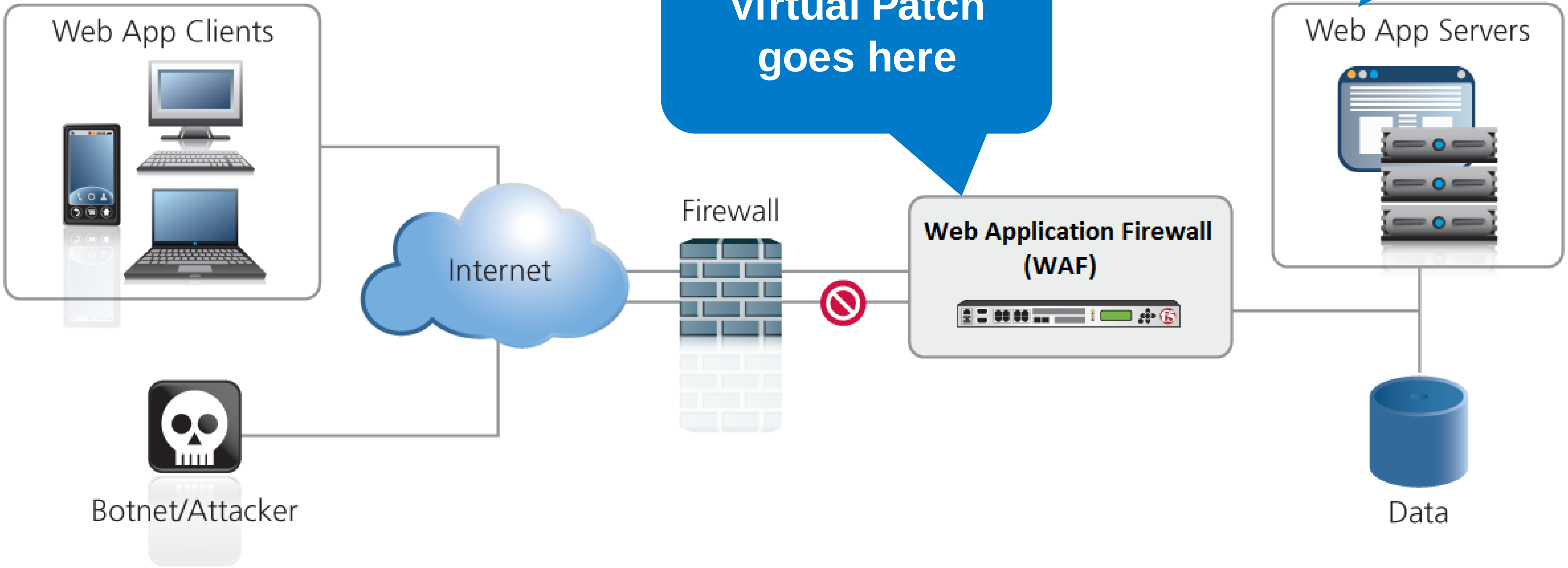
WAFs are a bit different

- They are ONLY for web applications and web services
- Securing vulnerable web applications is not easy for a product to deliver
- Impossible for a “jack of all protocols” security box

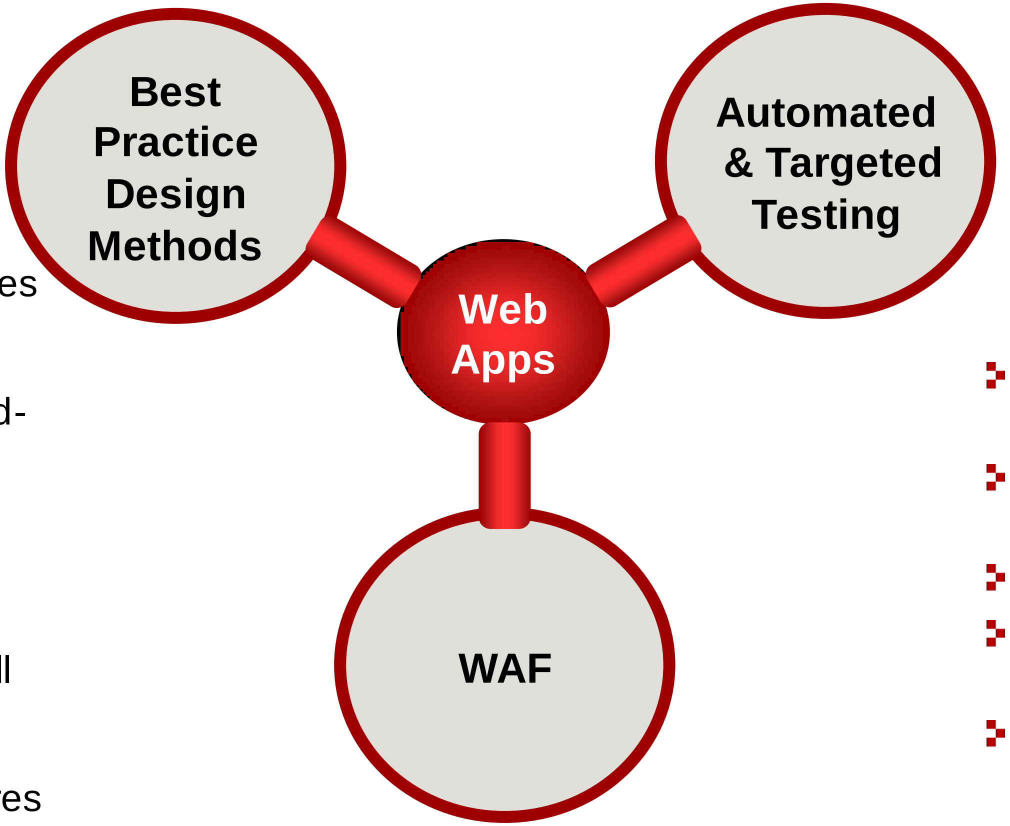
In front of the application:

Vulnerabilities Exploitable Here

Virtual Patch goes here



Application Protection Strategy

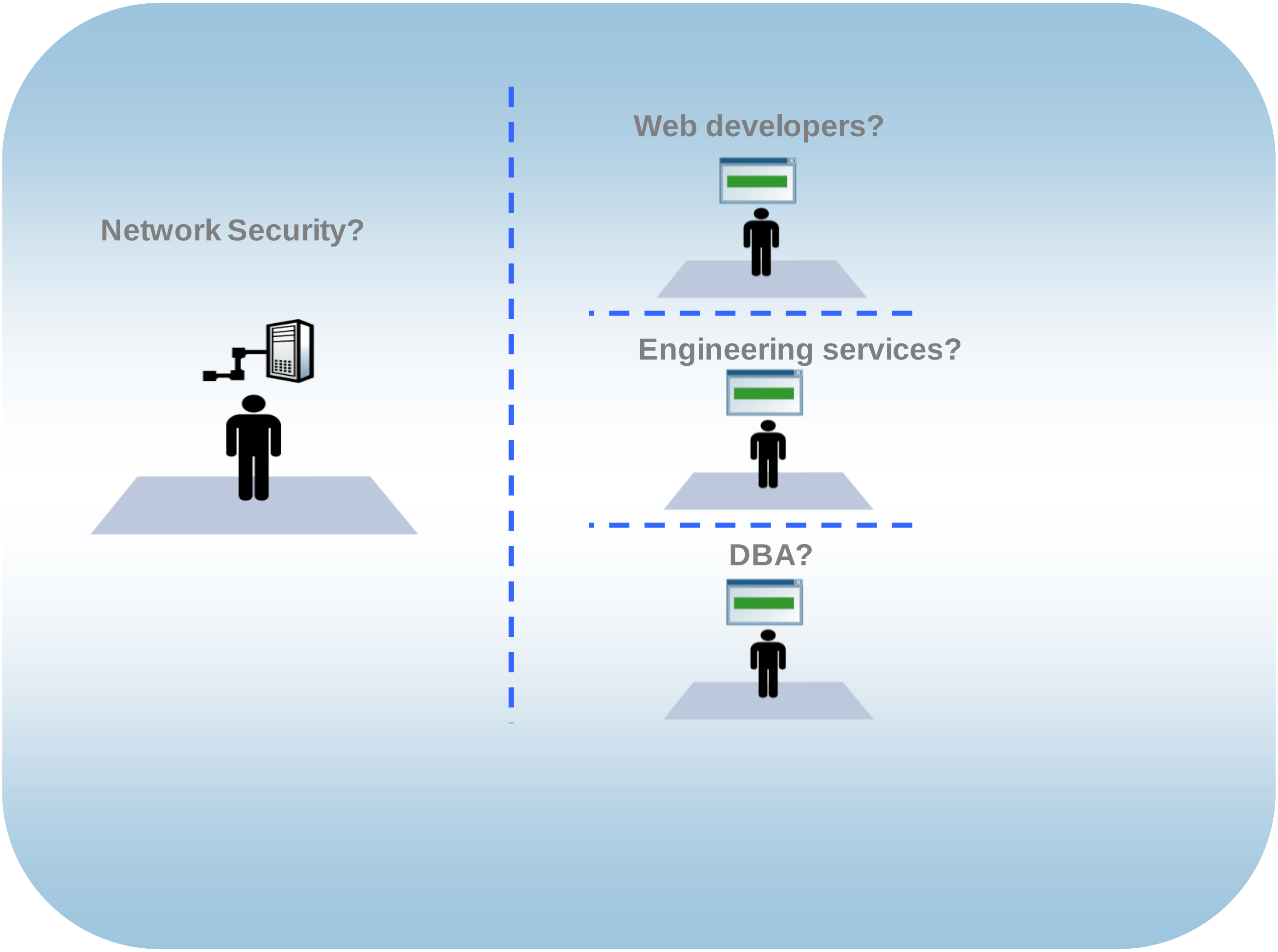


- Ideally there should be no vulnerabilities in the first place... However:
- Difficult to enforce; especially with third-party code
- Code changes may be a slow path to remediation, or impossible
- More secure coding requires more skill and time (cost)
- Some attack mitigation requires features developed within each application – expensive.

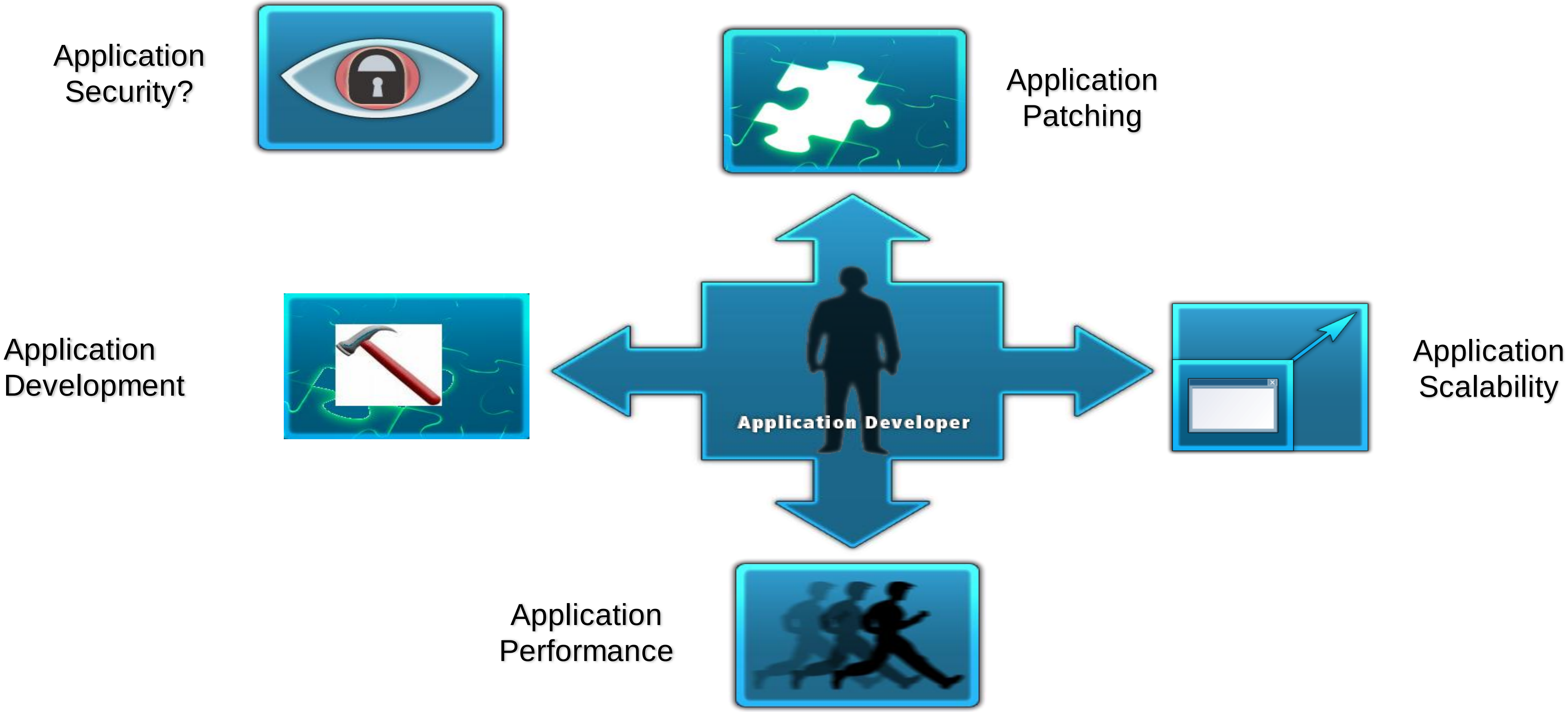
- Should be done regularly – ideally daily
- Scanning technology must be continually evolving
- Multiple tools gives greater coverage
- Operator skill the most important element
- Human penetration testing still required

- Toolkit to improve security – not silver bullet
- Provides remediation, protection, visibility
- Real-time 24 x 7 protection
- Management is important but need not be onerous
- Often the shortest path to remediation

Who is responsible for application security?



Developers are asked to do the impractical…



Application Security?

Application Patching

Application Development

Application Scalability

Application Performance

How long to resolve a vulnerability?

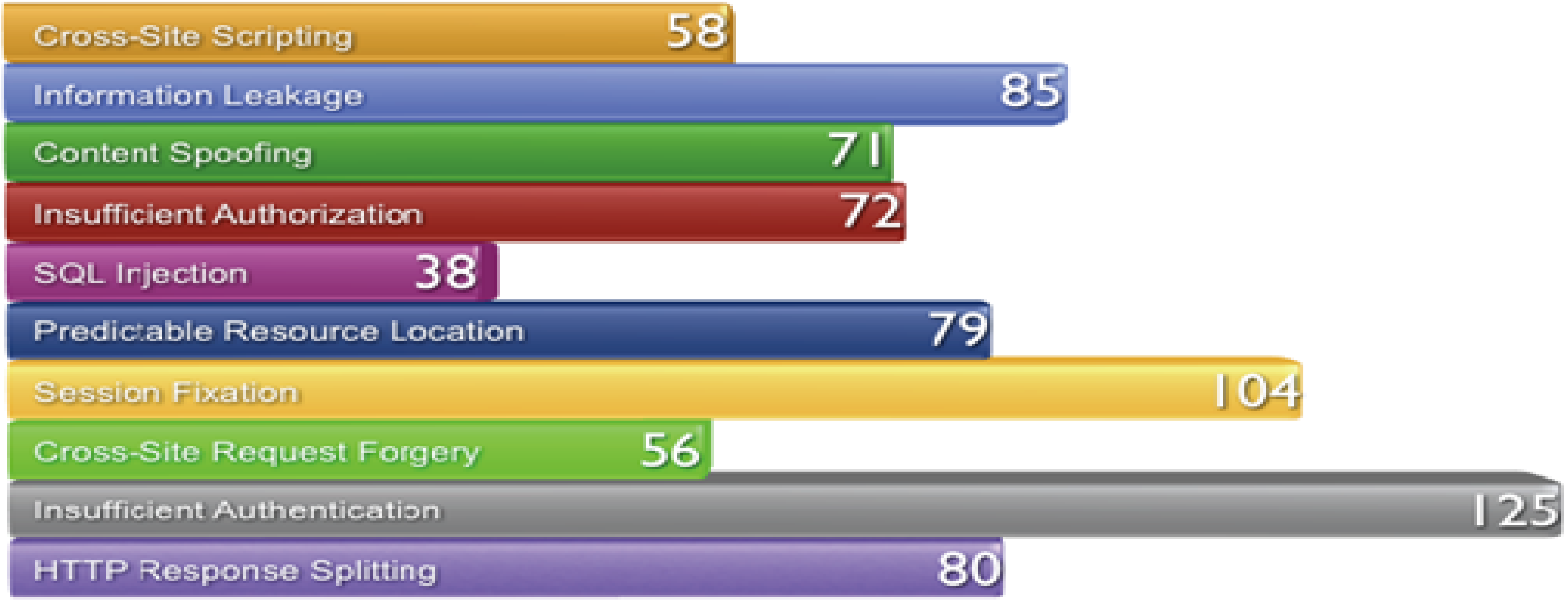


Figure 6. Average number of days for vulnerabilities to be resolved (sorted by class)

Website Security Statistics Report

Challenges of traditional network solutions (FW, IPS)

- HTTP attacks are valid requests
- HTTP is stateless, application is stateful
- Web applications are unique
  - there are no IPS signatures for YOUR web application
- Good protection has to have session context and awareness
- Encrypted traffic facilitates attacks…
- Organizations are living in the dark
  - missing tools to expose/log/report HTTP attacks

Why Not Fix the Code?

Sometimes:

- End of Life applications may not warrant the investment
- Third Party Code may not be available to fix
- Developers have moved on, organisation lacks the resource
- Platform and system dependencies prevent code fix or patch
- Developers asked to focus on new strategic initiatives
  - Patching old apps is sunk cost
  - Building new apps is business growth

> …From where I sit, we NEED WAFs to work, if nothing else but to provide development groups at least a few days of breathing room. I mean, consider the thousands of issues posted on sla.ckers.org, or XSSed.com… Is anyone really under the impression these will get fixed one at a time or anytime soon? And we're just talking about the XSS. What about the rest?
>
> \- Jeremiah Grossman

Pre-Conceived Perception

- No silver bullet
- Can always be bypassed by a skilled attacker
- No replacement for good code
- Only need one for PCI Compliance
    - Item 6.6 “Install a web-application firewall in front of public-facing web applications”

The Eye Opener

- Customer with very broken app (developed overseas)
  - Broken Auth
  - All data and feature restrictions on the client
  - All data validation on the client
- Advanced WAF able to “patch” all features

All of the Top 10?

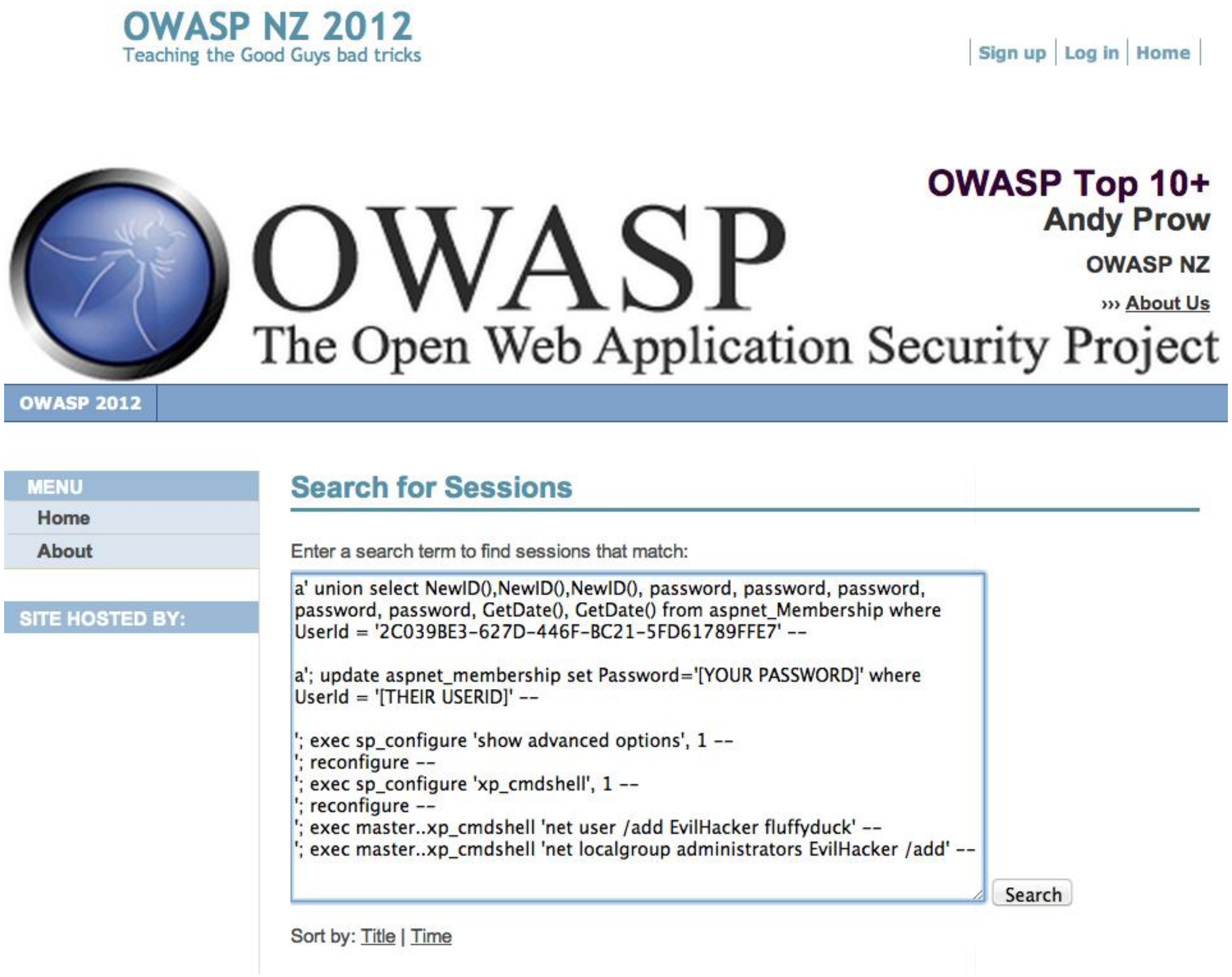
- Injection: SQL, OS & LDAP Injection
- XSS (Cross-site Scripting)
- Broken Auth. & Session Management
- Direct Object Reference
- XSRF (Cross-site Request Forgery)
- Security Misconfiguration
- Poor Crypto
- Unrestricted URL access
- Insufficient Transport Layer Protection
- Unvalidated Redirects and Forwards

The Easy Bits

- Injection: SQL, OS & LDAP Injection
- XSS (Cross-site Scripting)
- Direct Object Reference
- XSRF (Cross-site Request Forgery)
- Unrestricted URL access
- Insufficient Transport Layer Protection
- Unvalidated Redirects and Forwards

SQL Injection

SQL Injection



Search for Sessions

Enter a search term to find sessions that match:

```
a' union select NewID(),NewID(),NewID(), password, password, password,
password, password, GetDate(), GetDate() from aspnet_Membership where
UserId = '2C039BE3-627D-446F-BC21-5FD61789FFE7' --

a'; update aspnet_membership set Password='[YOUR PASSWORD]' where
UserId = '[THEIR USERID]' --

'; exec sp_configure 'show advanced options', 1 --
'; reconfigure --
'; exec sp_configure 'xp_cmdshell', 1 --
'; reconfigure --
'; exec master..xp_cmdshell 'net user /add EvilHacker fluffyduck' --
'; exec master..xp_cmdshell 'net localgroup administrators EvilHacker /add' --
```

Security Evasion using Encoding:

Basic SQL Injection via URI parameter:

```
' or 1=1 or '
```

Encoded version:

```
%27%20%6f%72%20%31%3d%31%20%6f%72%20%27
```

Evasion using Inline Comments:

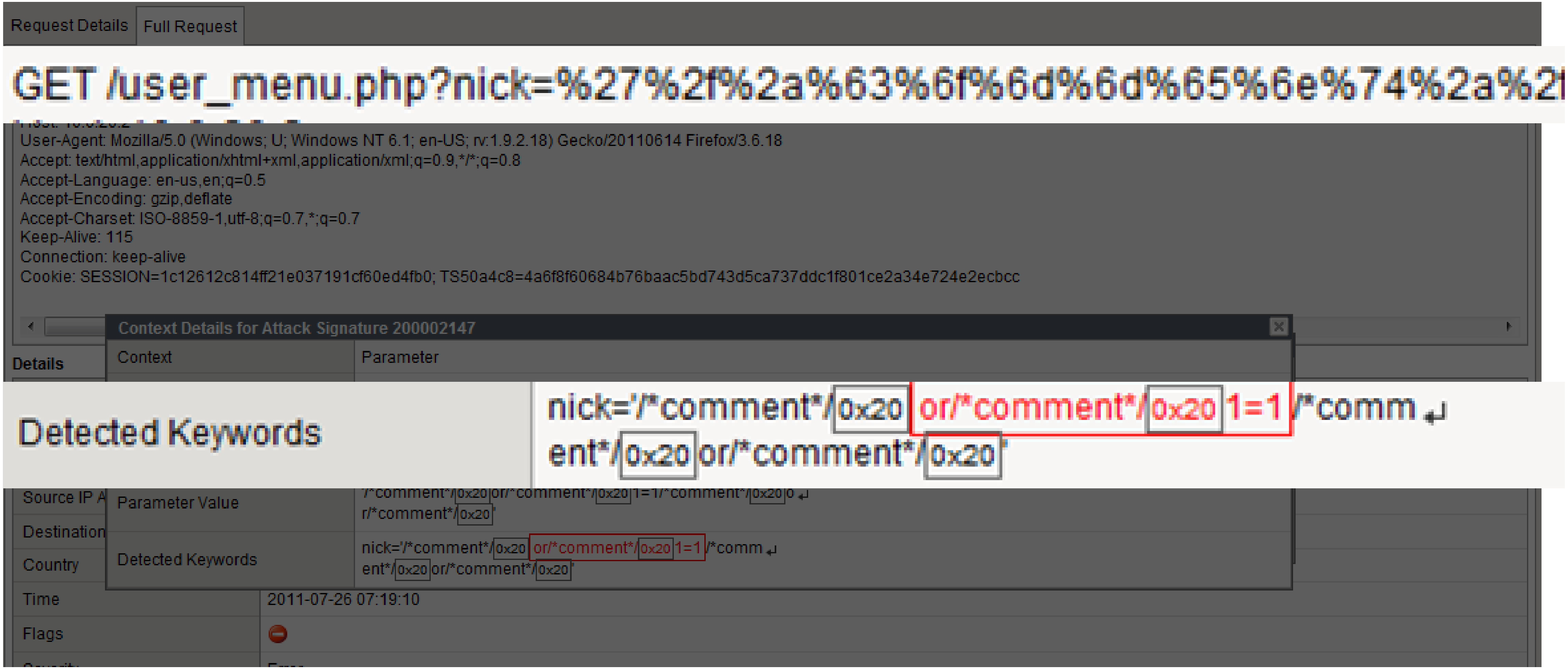
```
'/*comment*/ or/*comment*/ 1=1/*comment*/ or/*comment*/ '
```

Encoding and Commenting together:

Encoded, commented version:

```
%27%2f%2a%63%6f%6d%6d%65%6e%74%2a%2f%20%6f%72%2f%2a%63%6f%6d%6d%65%6e%74%2a%2f%20%31%3d%31%2f%2a%63%6f%6d%6d%65%6e%74%2a%2f%20%6f%72%2f%2a%63%6f%6d%6d%65%6e%74%2a%2f%20%27
```

Encoding and Commenting Together:



```
GET /user_menu.php?nick=%27%2f%2a%63%6f%6d%6d%65%6e%74%2a%2f
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.18) Gecko/20110614 Firefox/3.6.18
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
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Cookie: SESSION=1c12612c814ff21e037191cf60ed4fb0; TS50a4c8=4a6f8f60684b76baac5bd743d5ca737ddc1f801ce2a34e724e2ecbcc
```

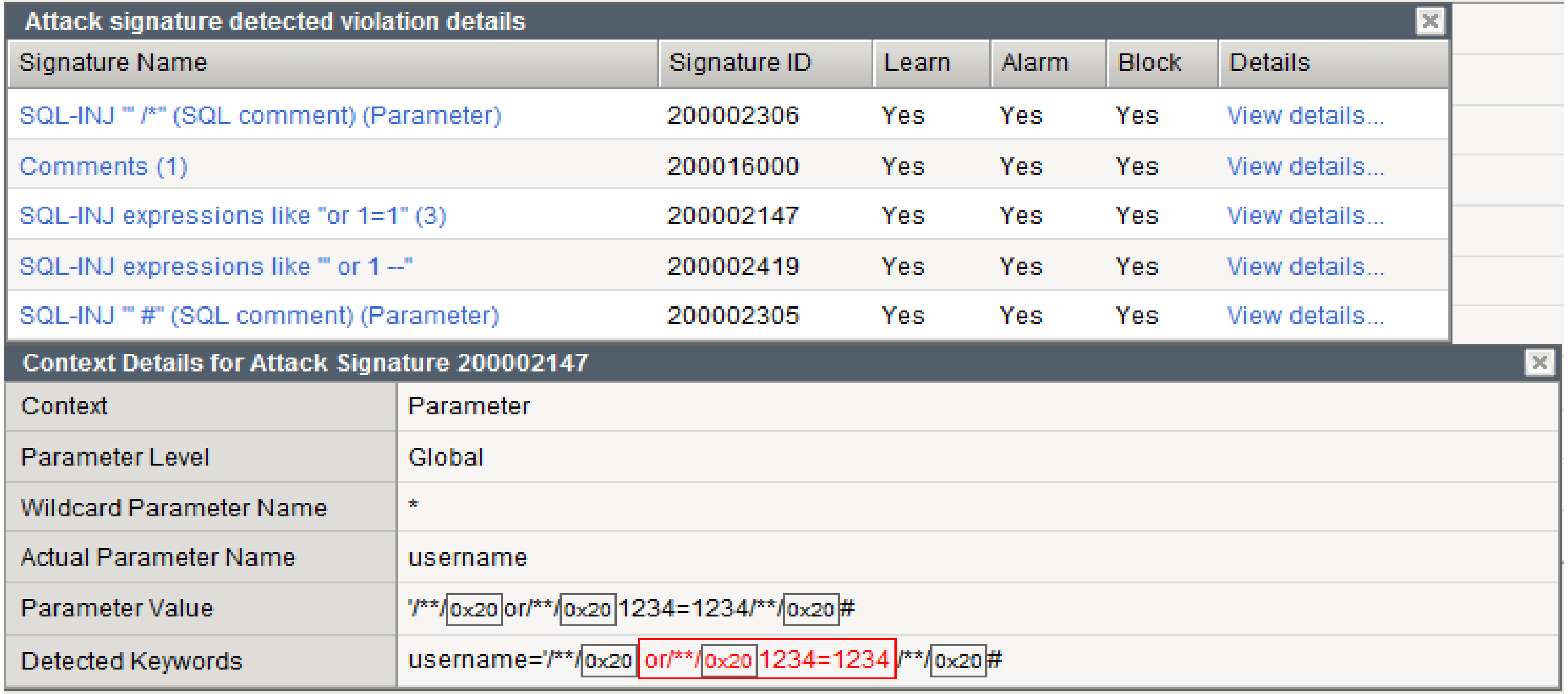
Context Details for Attack Signature 200002147

Context	Parameter
Detected Keywords	nick='/*comment*/0x20 or/*comment*/0x20 1=1 /*comm ↵ ent*/0x20 or/*comment*/0x20'
Parameter Value	'/*comment*/0x20or/*comment*/0x201=1/*comment*/0x20o ↵ r/*comment*/0x20'

Detected Keywords: nick='/*comment*/0x20 or/*comment*/0x20 1=1 /*comm ↵ ent*/0x20 or/*comment*/0x20'

Details	
Source IP A	
Destination	
Country	
Time	2011-07-26 07:19:10
Flags	⛔

Signature Matches on Decoded Request:



Attack signature detected violation details

Signature Name	Signature ID	Learn	Alarm	Block	Details
SQL-INJ "' /*" (SQL comment) (Parameter)	200002306	Yes	Yes	Yes	View details...
Comments (1)	200016000	Yes	Yes	Yes	View details...
SQL-INJ expressions like "or 1=1" (3)	200002147	Yes	Yes	Yes	View details...
SQL-INJ expressions like "' or 1 --"	200002419	Yes	Yes	Yes	View details...
SQL-INJ "' #" (SQL comment) (Parameter)	200002305	Yes	Yes	Yes	View details...

Context Details for Attack Signature 200002147

Context	Parameter
Parameter Level	Global
Wildcard Parameter Name	*
Actual Parameter Name	username
Parameter Value	'/**/0x20or/**/0x201234=1234/**/0x20#
Detected Keywords	username='/**/0x20or/**/0x201234=1234/**/0x20#

Not So Easy Bits…

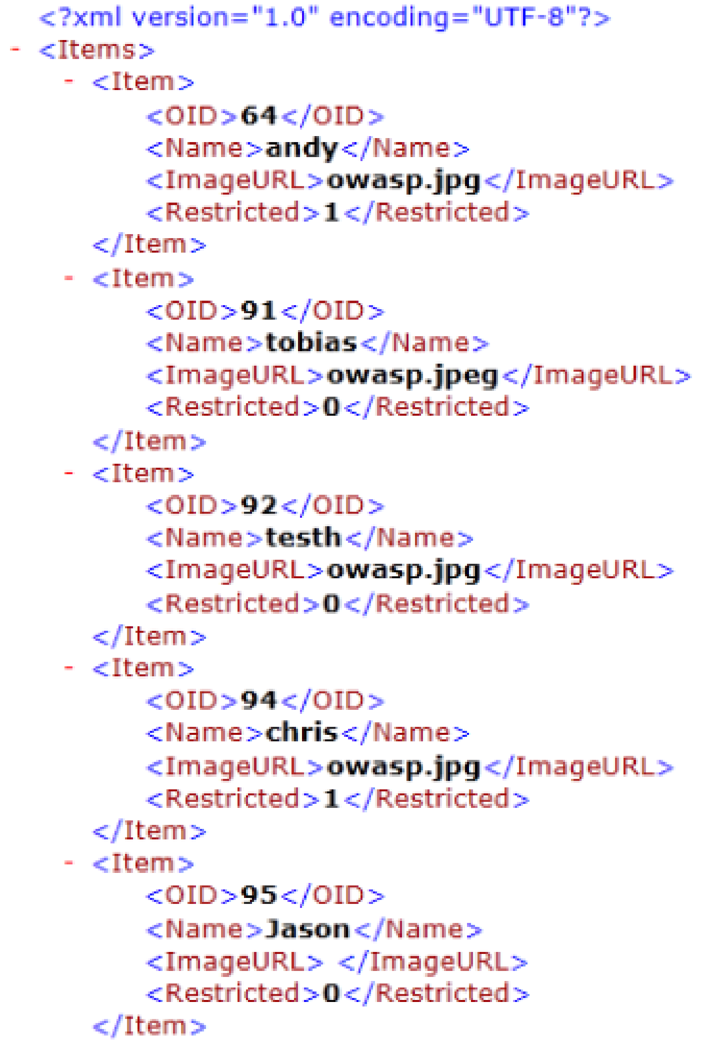
Not so Easy Bits…

- Broken Auth. & Session Management
- Security Misconfiguration – Exposed Web Services

- And Business Logic Flaws…

Authorisation – Data Acess

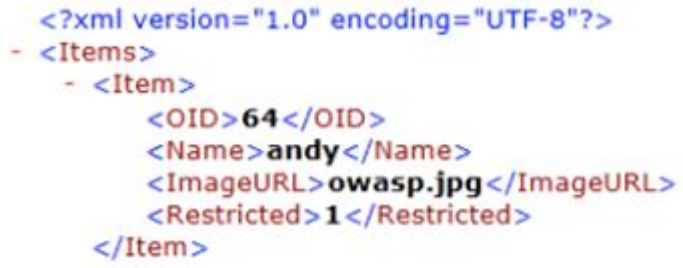
- All data is returned to the client app
- Client only shows restricted data if you’re allowed to see it…



```
  <?xml version="1.0" encoding="UTF-8"?>
- <Items>
  - <Item>
      <OID>64</OID>
      <Name>andy</Name>
      <ImageURL>owasp.jpg</ImageURL>
      <Restricted>1</Restricted>
    </Item>
  - <Item>
      <OID>91</OID>
      <Name>tobias</Name>
      <ImageURL>owasp.jpeg</ImageURL>
      <Restricted>0</Restricted>
    </Item>
  - <Item>
      <OID>92</OID>
      <Name>testh</Name>
      <ImageURL>owasp.jpg</ImageURL>
      <Restricted>0</Restricted>
    </Item>
  - <Item>
      <OID>94</OID>
      <Name>chris</Name>
      <ImageURL>owasp.jpg</ImageURL>
      <Restricted>1</Restricted>
    </Item>
  - <Item>
      <OID>95</OID>
      <Name>Jason</Name>
      <ImageURL> </ImageURL>
      <Restricted>0</Restricted>
    </Item>
```

Server Response Scrubbing

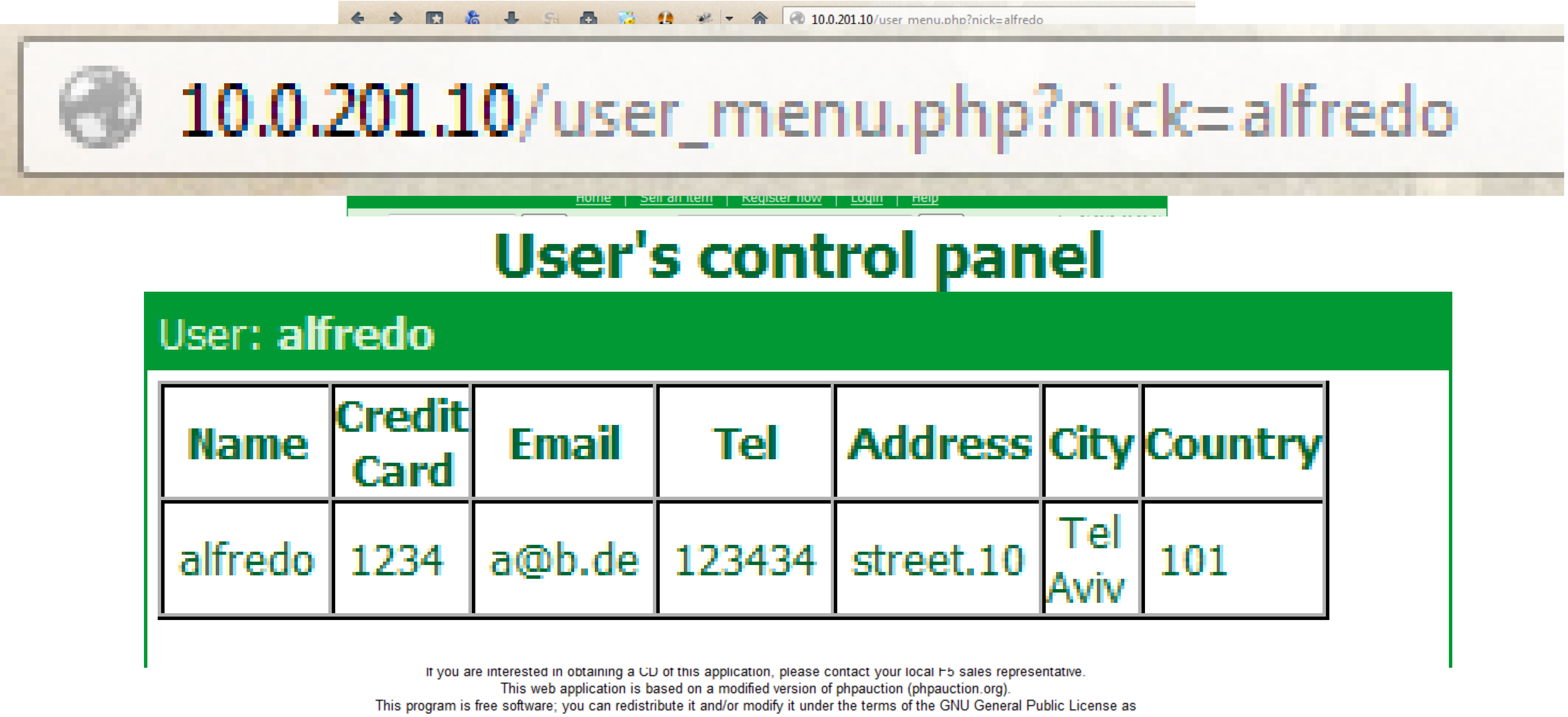
- Parse outgoing data set
- Match user identity and group with content
- Remove unauthorised Records from XML
- Return only authorised data



```
 <?xml version="1.0" encoding="UTF-8"?>
- <Items>
  - <Item>
      <OID>64</OID>
      <Name>andy</Name>
      <ImageURL>owasp.jpg</ImageURL>
      <Restricted>1</Restricted>
    </Item>
```

Broken Auth and Session Management

Log In as One User…



10.0.201.10/user_menu.php?nick=alfredo

10.0.201.10/user_menu.php?nick=alfredo

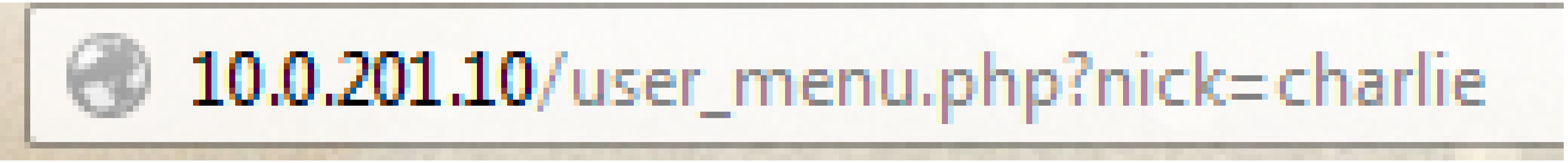
User's control panel

User: alfredo

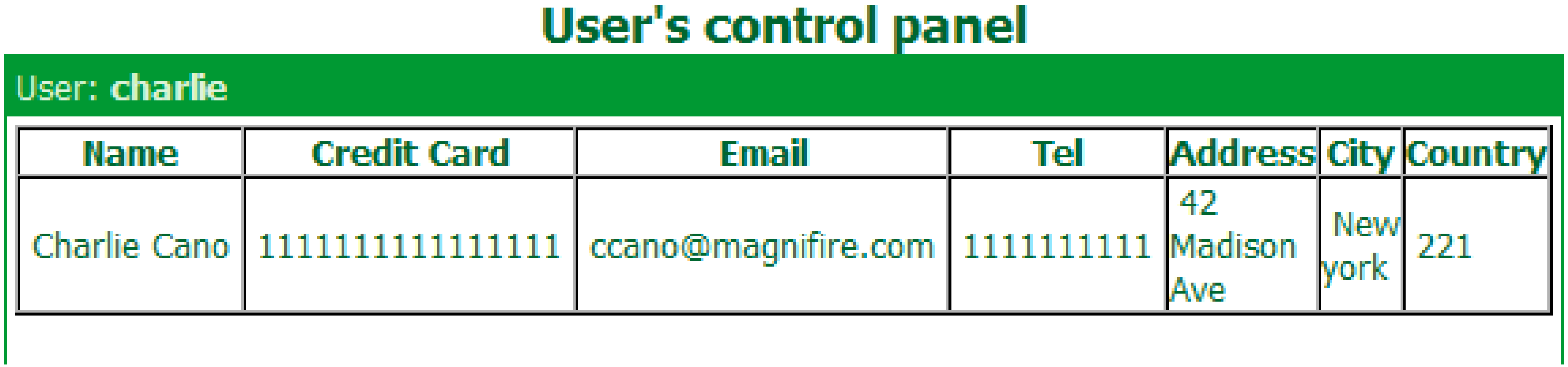
Name	Credit Card	Email	Tel	Address	City	Country
alfredo	1234	a@b.de	123434	street.10	Tel Aviv	101

If you are interested in obtaining a CD of this application, please contact your local F5 sales representative.
This web application is based on a modified version of phpauction (phpauction.org).
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as

View Another User’s Data:



10.0.201.10/user_menu.php?nick=charlie

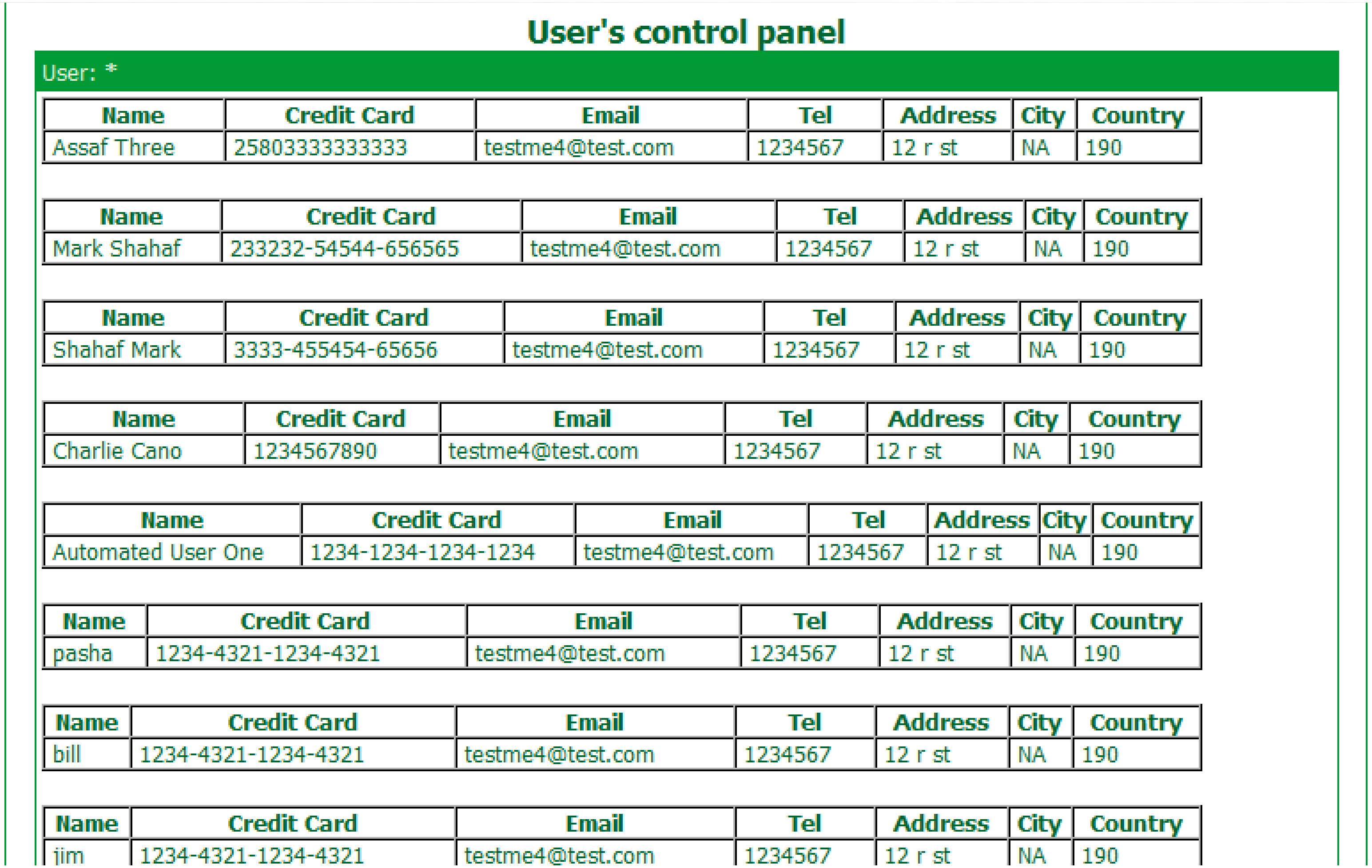


User's control panel

User: **charlie**

Name	Credit Card	Email	Tel	Address	City	Country
Charlie Cano	1111111111111111	ccano@magnifire.com	1111111111	42 Madison Ave	New york	221

View Everyone’s Data:



User's control panel

User: *

Name	Credit Card	Email	Tel	Address	City	Country
Assaf Three	25803333333333	testme4@test.com	1234567	12 r st	NA	190

Name	Credit Card	Email	Tel	Address	City	Country
Mark Shahaf	233232-54544-656565	testme4@test.com	1234567	12 r st	NA	190

Name	Credit Card	Email	Tel	Address	City	Country
Shahaf Mark	3333-455454-65656	testme4@test.com	1234567	12 r st	NA	190

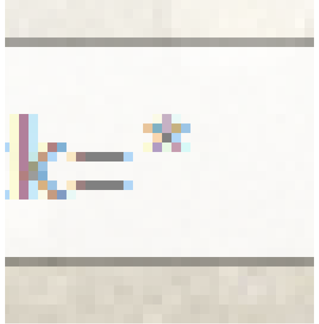
Name	Credit Card	Email	Tel	Address	City	Country
Charlie Cano	1234567890	testme4@test.com	1234567	12 r st	NA	190

Name	Credit Card	Email	Tel	Address	City	Country
Automated User One	1234-1234-1234-1234	testme4@test.com	1234567	12 r st	NA	190

Name	Credit Card	Email	Tel	Address	City	Country
pasha	1234-4321-1234-4321	testme4@test.com	1234567	12 r st	NA	190

Name	Credit Card	Email	Tel	Address	City	Country
bill	1234-4321-1234-4321	testme4@test.com	1234567	12 r st	NA	190

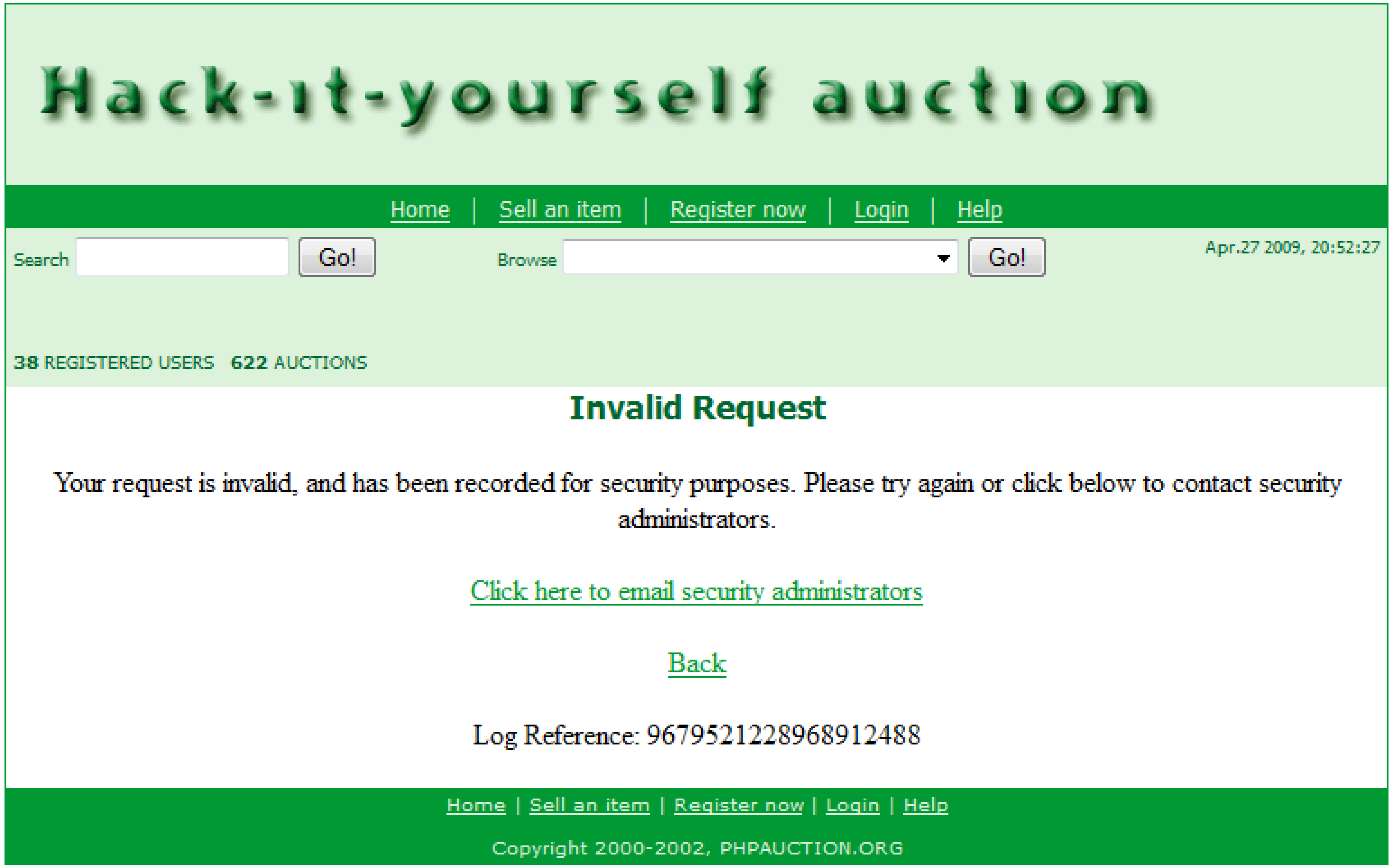
Name	Credit Card	Email	Tel	Address	City	Country
jim	1234-4321-1234-4321	testme4@test.com	1234567	12 r st	NA	190



Dynamic Parameter

- Server sends out parameters
  - Form fields, URI parameters in links, Cookies, etc
- WAF will parse and sign these in a cookie
- Inbound requests must present valid signature
  - Any value is OK, as long as it is YOUR value
  - Server must have supplied the parameter value within your session
  - Can’t be changed on the client side

Blocking Response



Hack-it-yourself auction
Home | Sell an item | Register now | Login | Help
Search
Go!
Browse
Go!
Apr.27 2009, 20:52:27
38 REGISTERED USERS 622 AUCTIONS

Invalid Request

Your request is invalid, and has been recorded for security purposes. Please try again or click below to contact security administrators.

Click here to email security administrators

Back

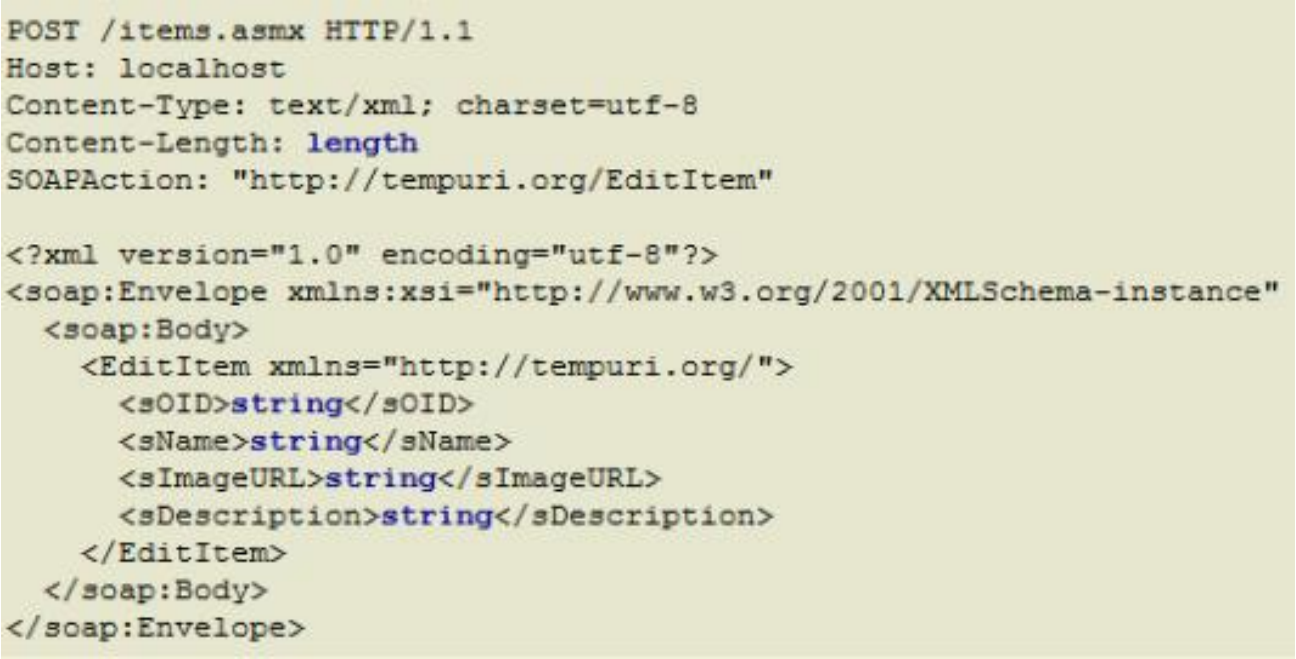
Log Reference: 9679521228968912488

Home | Sell an item | Register now | Login | Help
Copyright 2000-2002, PHPAUCTION.ORG

Exposed Web Services

Unauthorised Method Access

- App relies on Client side validation
- Back end methods all open

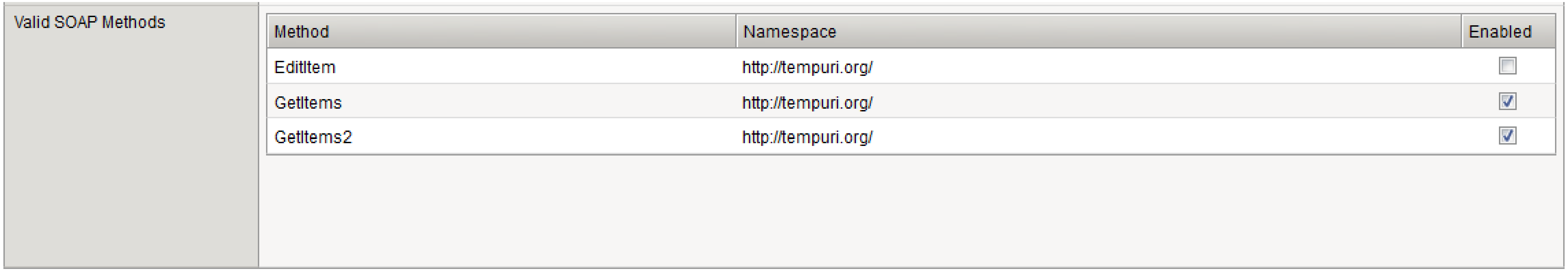


```
POST /items.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/EditItem"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <soap:Body>
    <EditItem xmlns="http://tempuri.org/">
      <sOID>string</sOID>
      <sName>string</sName>
      <sImageURL>string</sImageURL>
      <sDescription>string</sDescription>
    </EditItem>
  </soap:Body>
</soap:Envelope>
```

Authorisation for Method Access

- XML Firewalls provide this function
- Client Identity and Role may be used to disallow Method Access
- VLAN or IP address, ID, Device type, etc



Valid SOAP Methods

Method	Namespace	Enabled
EditItem	http://tempuri.org/	□
GetItems	http://tempuri.org/	☑
GetItems2	http://tempuri.org/	☑

Business Logic Flaws

Advanced Mitigation

- Authentication and Authorisation Wrapper
  - Auth proxy
  - 2 factor
  - Certificate, Kerberos, Forms based, NTLM, etc
- Response Modification
  - EXIF tag XSS example
  - CSRF token example
- Enforcing Order of Events (“Flow”)
- Full request and response parsing and modification
  - Session awareness – with session principles
  - Programmable framework used to mitigate app-specific cases

Responsive Actions:

- Drop Request
- Log, Email, SNMP trap
- Respond with Blocking content
  - HTML – Security warning
  - Link to email administrators in case of issues
  - SOAP Fault for web services
  - Javascript injection for AJAX
  - Honeypot silent redirect
- Query the client a bit further
  - Browser or Robot?
  - Send back Javascript to test client before trusting session
- Your ideas here...?

Questions...

devcentral.f5.com

facebook.com/f5networksinc

linkedin.com/companies/f5-networks

twitter.com/f5networks

youtube.com/f5networksinc