SANS Technology Institute

STI Group Discussion/Written Project

Data Retention and Cost Effective Data Loss Prevention Techniques

Version 1.1

Eric Conrad

Mason Pokladnik

Manuel Santander

08

Table of Contents

1. Executive Summary

GIAC Enterprises has reached a critical juncture of growth. The company has grown organically, to become the largest supplier of fortune cookie fortunes in the world. The company has experienced "growing pains" in regards to information security and overall information technology. Informal IT processes are beginning to break down due to increased load and complexity.

Information security is of the highest concern, as the company's primary asset is proprietary information in the form of small strings (the fortunes), and the loss of this information would be devastating to the company. GIAC Enterprises must implement a data loss prevention (DLP) plan, including formal security policies and robust security architecture, to address the risk to its proprietary information.

We recommend a three-phase approach for mitigating this risk. Phase 1 will include a complete assessment of proprietary data, both at rest and in transit. Phase 2 will include a short-term focused remediation effort, including detecting proprietary information in policy-violating situations and removing any unnecessary or outdated information. Phase 2 also includes making a number of infrastructure and policy changes and enhancements to protect remaining proprietary data. Major architectural changes include the implementation of a new web application firewall, and multiple new mechanisms for detecting proprietary data in transit and at rest. Phase 3 will include longer-term enhancements to the GIAC Enterprise security infrastructure.

Due to the critical nature of this threat, and the compressed timeline due to a potential acquisition, we have created a project plan that will complete phases 1 and 2 within 15 days, requiring three engineers, totaling 45 staff days. We are recommending open-source tools for the majority of the remediation effort due to the power and flexibility of the tools and due to the projects' cost constraints.

2. GIAC Enterprises' Data Retention Policy Recommendations

We recommend updating the Data Retention Policy to reduce the amount of information that is available for an attacker to steal from our fortune cookie sales infrastructure. All of the technical countermeasures discussed later in this document will be greatly enhanced if the information attackers are pursuing is unavailable to be stolen. Incidents such as the TJ Maxx[1] credit card theft show that organizations not taking steps to limit the amount of sensitive information available are setting themselves up for massive and expensive data losses. In order to reduce this threat, we recommend that management not only look at where information can be stored and transmitted, as we are analyzing with this data loss prevention project, but also look for ways to remove sensitive data that is no longer necessary for day to day operations.

Such sensitive information could include:

- Credit card information – Requirement 3.1 of the PCI Data Security Standard requires that information such as the credit card number, card verification code and billing zip code are retained no longer than is necessary for business or legal reasons.[2] The same protections should apply to debit cards or any other forms of payment.

- Personally-Identifiable Information (PII) on customers and contractors – Our current retention policy states that information about customers and contractors could be kept for years after they have stopped any transactions with the organization. We recommend that usernames, password hashes/reset questions, site browsing and purchase history, addresses and any other unique information be reviewed for possible removal from the fortune cookie sales system when no longer in use.

- Backups – The retention policy does not specifically deal with the issue of encrypting sensitive information so that it cannot be stolen from the backups. There are multiple approaches to this problem. The fastest being to encrypt the entire backup as it is written to tape. The safer long-term strategy is to encrypt the records wherever they are stored; therefore, they are encrypted whenever they are backed up.

2.1.1. Data Sensitivity Labels

GIAC Enterprises has already established a Data Labeling Policy[3] that will be a great aid in moving forward. The policy will allow us to identify immediately any properly labeled content in the event it leaves our network. However, not all information is labeled and will therefore have to be treated as business proprietary information in compliance with the Data Labeling Policy.

The following strings are used as labels in GIAC Enterprises data classification[4]

- *GIAC Enterprises Public*
- *GIAC Enterprises HIPAA*
- *GIAC Enterprises HR*
- *GIAC Enterprises Business Proprietary*
- *GIAC Enterprises Technical Proprietary*

3. Technical Countermeasures for the Network and Endpoints

Before the previous CISO resigned, she recommended that we purchase data loss prevention software from Vontu (now owned by Symantec). The Vontu product has several impressive features to identify corporate information across disparate data stores throughout the enterprise including databases, email systems and regular file systems.[5] Our charge was to replicate as many of these features as we could at a greatly reduced acquisition cost and a timeframe of 15 days.

3.1. Defense in Depth

GIAC Enterprises will employ defense-in-depth to protect its proprietary information. This includes a robust security architecture, server hardening, and policy enforcement. The defense also includes a number of detective methods, including proprietary information detection, as well as intrusion detection.

3.2. Canary Cookies

A “canary trap” is a technique of disseminating information to multiple sources each with a unique signature so that if that information is leaked the source of the leak can be identified.[6] We propose a modified version of this technique where certain strings will be inserted into the database such that they look like normal records, but should never be returned during the normal operation of the program.

We will call the canary trap fortunes ‘canary cookies.’ These fortunes are not real, and when detected anywhere but in the database, various parts of the security architecture will send an alert. An example canary cookie is “C is for Cookie, and thats good enough for me.”[†] We will attempt to track this cookie within GIAC Enterprises, both in transit and at rest.

3.3. Leveraging Existing Systems

One option available to us is to use the content analyzer feature of the IBM Proventia appliance we purchased when implementing the data center. The device has been used primarily as a traditional firewall, a feature we intend to continue to leverage for our DLP architecture.

The Proventia is also capable of analyzing multiple protocols including HTTP, SMTP and several instant messaging protocols.[7] It can also look inside compressed files and certain document formats. Using the built-in and up to 8 custom signatures, we could add another layer of coverage that would address more than just traffic headed to our fortune cookie application. Given the processor load this place on this Firewall/IPS/VPN gateway all-in-one machine we strongly suggest using highly targeted searches such as credit card numbers and matches for our ‘canary cookie’ trap strings. Specific example regular expressions are detailed below.

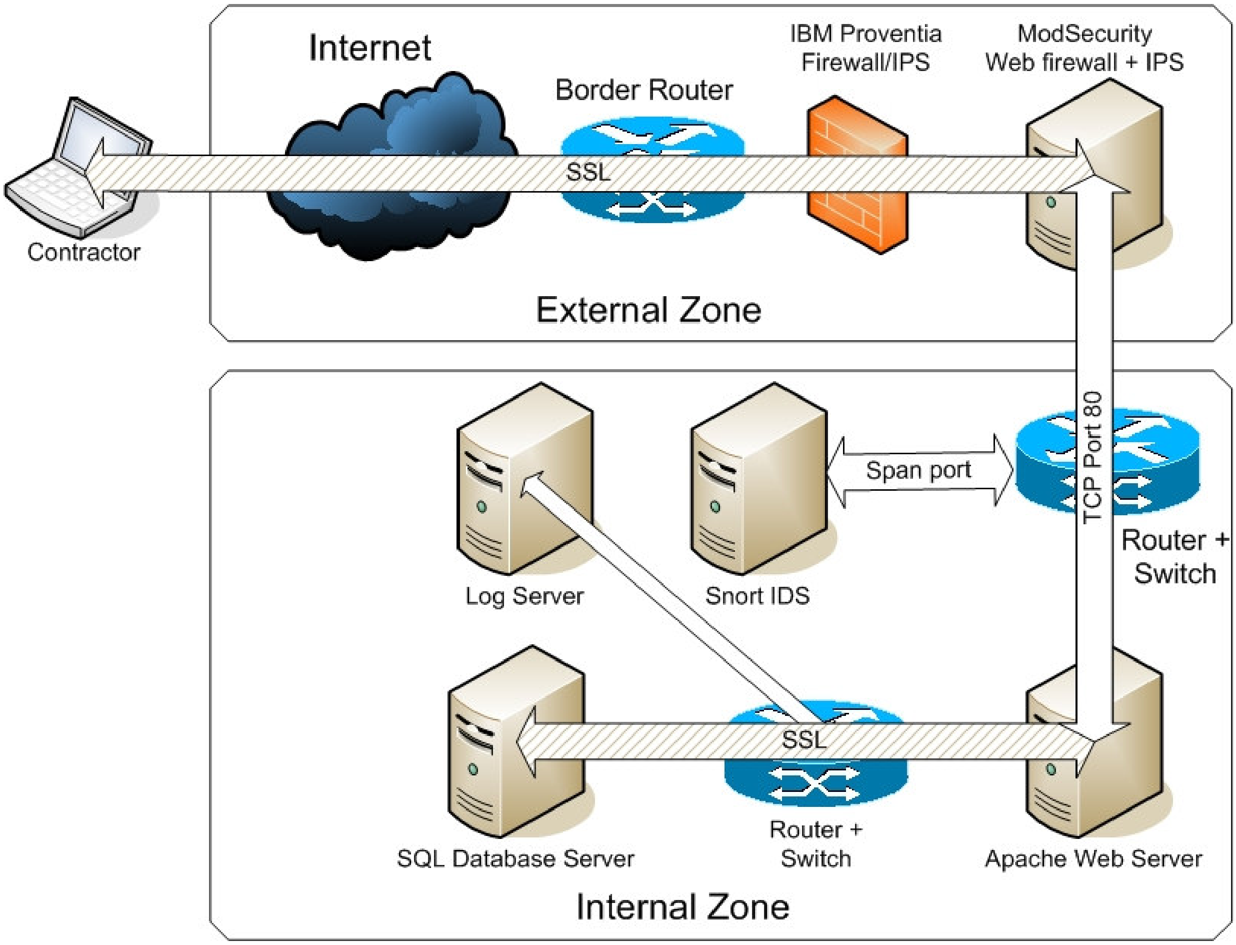
3.4. Network-Based Protection

3.4.1. GIAC Enterprises Architecture

GIAC Enterprises provides a variety of Internet services, including fortune submission by contractors around the world. While studying the current network from a DLP perspective, we have identified several upgrades that will help us to identify when our intellectual property is under attack.

[†] Apostrophe was left from ‘thats’ intentionally; it makes the string more unique, and it avoids requiring an escape for the apostrophe inside a search string.

This diagram shows the proposed logical 'fortune' architecture:[‡]



Contractors use a browser-based application to submit new fortunes. 128-bit SSL encryption is used. Both client and server certificates are used, providing mutual authentication of client systems as well as the server. This will necessitate the creation of a new enterprise root certificate authority (CA) to issue the client certificates. This will be a standalone CA that will not be accessible from the Internet.

A border router and internet firewall protect internet-facing systems, including a server running Modsecurity in reverse proxy mode. SSL sessions terminate on the Modsecurity server. Modsecurity provides application-level intrusion prevention services (dropping detected malicious traffic), as well as traffic normalization.

[‡] This is a simplified logical diagram: certain features (like redundancy for high availability) are omitted.

The traffic then passes (unencrypted) through another router with an embedded switch blade to an Apache server running the fortune cookie submission service. The switch blade contains a span port with an attached Snort Intrusion Detection System.

The Apache server queries a SQL database on another network, via SSL.

All internal routers include a stateful packet inspection engine, and restrict traffic to allow only required hosts and services. Should an attacker breach the Modsecurity server from the internet, as one example, the next internal router will only allow access to http (TCP port 80) to the Apache server, and syslog (TCP/UDP port 514) to the central logging server.

A centralized syslog-ng server will be used for centralized logging and event correlation. Hosts should use a TCP connection to the log server if supported.

3.4.2. Simplified Firewall/Router ACLs

Here is a simplified list of ACLs for the firewall and relevant routers:

Source	Destination System	Destination Port	Action
any	Modsecurity	TCP 443	Allow
Modsecurity	Apache	TCP 80	Allow
Apache	MYSQL	TCP 3306	Allow
internal_network	Syslog-ng Server	TCP/UDP 514	Allow
any	Any	any	Deny

3.4.3. Regular Expressions to Detect Sensitive Data

Regular Expressions (regex for short) are a powerful mechanism for matching text strings. GIAC Enterprises will leverage the power of regexes to detect sensitive data.

This regex will detect the canary cookie:

```
/C is for Cookie, and thats good enough for me/
```

The following regex will identify the sensitive GIAC Data protection labels:

```
/GIAC Enterprises (HIPAA|HR|(Business|Technical) Proprietary)/
```

The 'GIAC Enterprises Public' label is not matched, as information with that label is not proprietary.

The following regex will detect credit card numbers.[§]

```
/ (6011|5[1-5]\d{2}|4\d{3}|3\d{3}) \d{4} \d{4} \d{4}/

/ (6011|5[1-5]\d{2}|4\d{3}|3\d{3})-\d{4}-\d{4}-\d{4}/

/ (6011|5[1-5]\d{2}|4\d{3}|3\d{3})\d{12} /
```

[§] Based on Bleeding Snort Policy Rules, http://www.bleedingthreats.net/rules/bleeding-policy.rules

```
/ (3[4|7]\d{2}|2014|2149|2131|1800)\d{11} /

/ (3[4|7]\d{2}|2014|2149|2131|1800) \d{4} \d{4} \d{3} /

/ (3[4|7]\d{2}|2014|2149|2131|1800)-\d{4}-\d{4}-\d{3} /

/ (30[0-5]\d|36\d{2}|38\d{2})\d{10} /

/ (30[0-5]\d|36\d{2}|38\d{2}) \d{4} \d{4} \d{2} /

/ (30[0-5]\d|36\d{2}|38\d{2})-\d{4}-\d{4}-\d{2} /
```

These regular expressions may result in false positives, when an alert is generated for non-credit card data. The team handling these alerts must keep this in mind during their incident handling process. More specific approaches can be developed during phase 3 to lower false positives.

3.4.4. Snort Canary Cookie Detection

The Snort Intrusion Detection[8] system is a powerful open-source IDS which will make an excellent addition to GIAC Enterprises' Poor-man's DLP architecture. It supports detection via regular expressions, among many other features.

Due to the complexity and potentially high total cost of ownership required for a Snort sensor with lots of rules, GIAC Enterprises will deploy a small amount of high-value rules. These rules will be designed to detect proprietary information, such as the canary cookie.

This Snort rule will detect the canary cookie moving on the network in any direction:[**]

```
alert ip any any -> any any (msg:”Canary Cookie 1”; content:"C is for
Cookie, and thats good enough for me"; rev:1)
```

This Snort rule will detect credit card numbers flowing to external networks:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Credit Card
number!"; flow: from_server,established; pcre:"/(6011|5[1-
5]\d{2}|4\d{3}|3\d{3}) \d{4} \d{4} \d{4}/i"; rev:1;)
```

3.4.5. Syslog-ng Server

A Unix-based Syslog-ng server will be deployed during phases 1 and 2. Due to budget and time constraints, we plan to write a small amount of focused detections scripts in the Perl scripting language. These scripts will detect any logs showing Canary Cookies (such as ClamAV alerts), credit card number detection, etc.

As budget allows, GIAC Enterprises should consider deploying a more formal log correlation device during phase 3, such as Tenable Security's LCE. [9]

[**] Canary cookies should never move, so the rule will alert on all ports, in all directions. Due to the criticality of this alert, the performance cost of 'any source, any destination, any ports' is considered worthwhile.

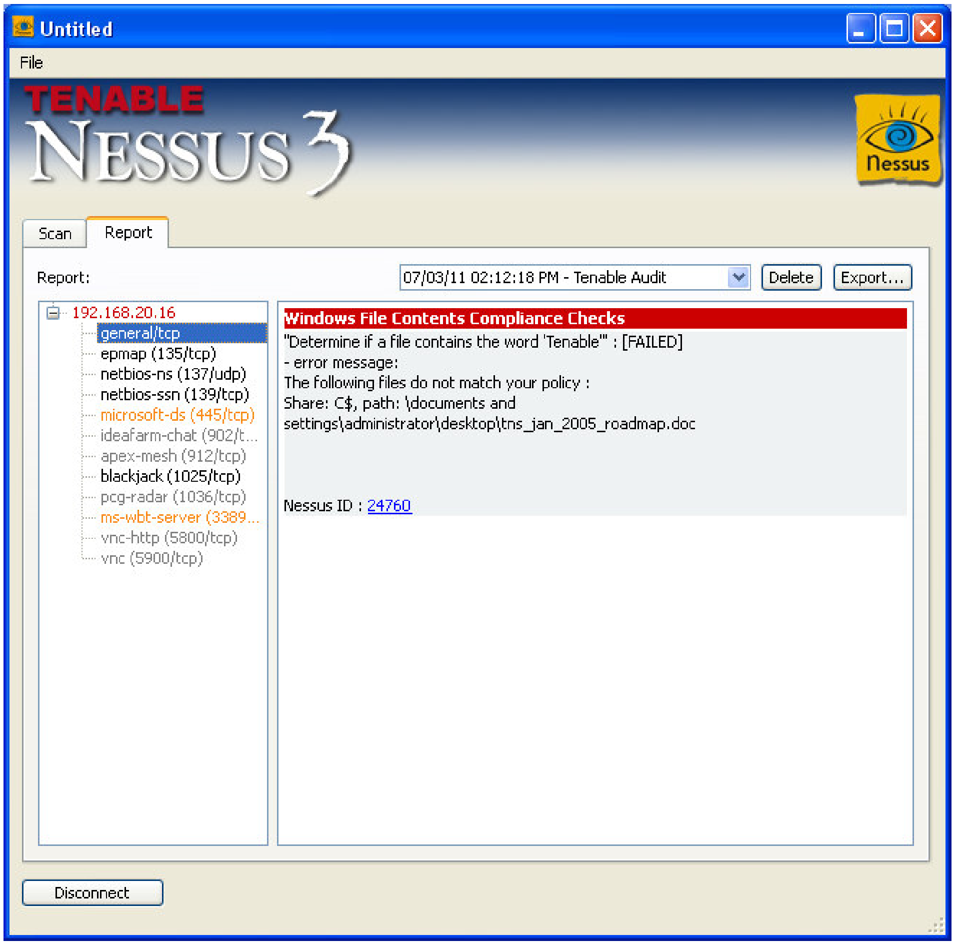
3.5. Endpoint Protection

3.5.1. Nessus

Nessus is a security vulnerability scanner able to perform security assessments on computers, servers or any other device connected to the network. It has several plug-ins that can be useful for detecting content in files for Windows and Unix systems.[10]

No source code is available for the commercial variant, but it is free to use on a private network to assess internal vulnerabilities.

Plug-in 24760, named Windows File Contents Compliance Checks, is able to detect sensitive data on Windows computers based on regular expressions. If a match is found, the file names will be shown with full location showing that the policy assessment failed on the computer.



```
Windows File Contents Compliance Checks
"Determine if a file contains the word 'Tenable'" : [FAILED]
- error message:
The following files do not match your policy :
Share: C$, path: \documents and
settings\administrator\desktop\tns_jan_2005_roadmap.doc


Nessus ID : 24760
```

If no match is found, the policy compliance shows as passed. This can be done for all internal computers.

3.5.2. ClamAV

ClamAV will be used as a “Poor Man’s DLP Agent,” running on client systems within GIAC Enterprises. All ClamAV instances will be configured to log centrally via syslog. The central syslog-ng server will be configured with scripts designed to alert when strings like ‘CanaryCookie1’ (see below) are detected.

Write custom ClamAV signature to detect the canary cookie:

```
# echo “C is for Cookie, and thats good enough for me”|
/usr/local/bin/sigtool --hex-dump
```

```
4320697320666f7220436f6f6b69652c20616e6420746861747320676f6f64206
56e6f75676820666f72206d650a
```

Create the ClamAV signature:

```
CanaryCookie1:0:*:4320697320666f7220436f6f6b69652c20616e642074686
1747320676f6f6420656e6f75676820666f72206d650a
```
[††]

In addition to our custom signatures, the maintainers of ClamAV are also working on a DLP module that we may able to leverage in the future.[11]

3.5.3. ‘find’ Command

‘find’ is a command available on Unix and Windows systems with Cygwin installed. It is able to search a filesystem for specific files, and execute commands on those files (among many other features). We will use find along with egrep (extended grep, which may search files for regular expressions).

Run ‘find’ command across Unix and Windows systems, looking for the bad cookie:

```
find . -type f -exec egrep 'C is for Cookie, and thats good
enough for me' {} \;
```

‘find’ may also be used to detect GIAC data labels and a specific credit card number format (see above) respectively:

```
find . -type f -exec egrep 'GIAC Enterprises
(HIPAA|HR|(Business|Technical) Proprietary)' {} \;
```

```
find . -type f -exec egrep '(6011|5[1-5]\d{2}|4\d{3}|3\d{3}) \d{4}
\d{4} \d{4}/' {} \;
```

[††] Colon delimited fields are MalwareName:TargetType:Offset:HexSignature. ‘0’ for a TargetType means any file. ‘*’ for an offset means any offset. See: http://www.clamav.net/doc/latest/signatures.pdf

4. Database Security

Database security to prevent Data Loss does not rely on the database by itself but to the whole architecture. The following principles are required to minimize the risk of data loss directly from database:

4.1. SSL Encryption between Web Server and Database

SQL Logon Information normally travels across the wire in plaintext. It is possible to capture usernames and passwords to subvert the web server and steal information from the database. To fix this issue, the database support SSL encrypted connections from client to server. This requires the following changes to the way the clients connect:

- The same certificate authority used to issue client certificates for our secure website will be used to secure connections to the database. Openssl can do the required job by providing the Mysql database server with a CA Certificate and a server certificate.
- Recompile the Mysql database and client to allow SSL support (unless already supported)
- Initiate the server specifying the CA certificate, server certificate and server private key.
- Before calling the *mysql_real_connect()* function from the client, call the mysql_ssl_set() function to specify the CA certificate, client certificate, private key and cipher to use for the encryption.

4.2. Restricted Database Network Access

The database must be accessible only from the front-end Apache server to prevent direct connections and brute-force attacks directly against the database. This will also prevent the capturing of any password hashes, thereby preventing rainbow table (a pre-computed dictionary of encrypted passwords) attacks on the captured hashes.

4.3. Password Complexity Enforcement

Since there is no native complexity password checking at the database, a stored procedure must be written to supply this to avoid trivial passwords for database admin users like root. A regular expression for this could be:

```
/^.*(?=.{10,}) (?=.*[A-Z]) (?=.*[a-z]) (?=.*[@#$%^&+=]).*$/
```

This enforces the following rules:

- Password is at least 10 characters long
- The password must contain at least one special character from the following set: *@#$%^&+=*.
- The password must contain at least one lower letter, one capital letter and one special character.

4.4. Database Audit Trails

The new GIAC Enterprise security architecture should mitigate direct unauthorized attempts to break into the MySQL database and steal information. However, there might be flaws on the other security architecture components, such as a flaw on a regular expression for checking malicious input from users on the application front-end or an error on role assignment, allowing attackers additional privileges to manipulate the information in an apparently legal way. In that case, the security software or devices may not issue an alert. According to the Data Labeling Policy, any sensitive information must have an audit trail for the table, for a specific field, or for transactions or sessions.

4.5. MySQL Least Privilege Configuration

If the SQL Injection controls are misconfigured and an attack manages to bypass them, it's important to restrict the privileges of the database to execute actions inside the operating system via perl execution sentence or vulnerability issues on the database.

4.6. Stored Procedures

One non-trivial change we suggest attempting to implement is altering the fortune cookie application to use MySQL stored procedures. This is a major change to the application and will need to be tested thoroughly before it is placed in production. The benefit of the change is that the userid the application runs under will only have access to the stored procedures, not the underlying database tables themselves.[12] Additionally, the use of stored procedures allows us one final location to perform input validation before attempting to query or update the database. The combination of these factors should allow us to mitigate many potential SQL injection attack methods as incorrect variables passed to the stored procedure should return errors and any attempt to directly query an underlying table will fail for lack of permissions.

4.7. Web Application Firewall

After ensuring the database will not be accessed from other place of the network than the webserver and permissions are well set, the Modsecurity module installed on the apache web server will be the web application firewall for this application enforcing the prevention of data loss by the application with the following functionality:

4.7.1. Credit Card Number Detection

The ModSecurity program has a module to detect credit card data loss by the application by using the *verifycc* operator. The following rule will be configured:

```
SecRule ARGS "@verifyCC \d{13,16}" \
"phase:2,sanitiseMatched,log,auditlog,pass,msg:'Potential credit
card number data loss'"
```

4.7.2. GIAC Data Protection Labels

The following Modsecurity rule is able to detect the non public GIAC Protection Data Labels when sent over the wire:

```
SecRule ARGS “@rx GIAC Enterprises (HIPAA|HR|(Business|Technical)
Proprietary)”
```

4.8. Client and Server Side Validation

In a recent video posted to the Youtube.com website, Dr. Eric Cole makes an interesting point. If web applications perform **both** client-side data validation and server-side validation, when properly implemented the server-side validation should never need to be used.[13] Therefore, the server-side rules can be used as a form of attack detection. By adding client side validation to our fortune cookie application, as well as tracking if a server side rule is ever triggered we can detect people trying to bypass the normal interface. That IP address can then be temporarily blacklisted to prevent any further attempts.

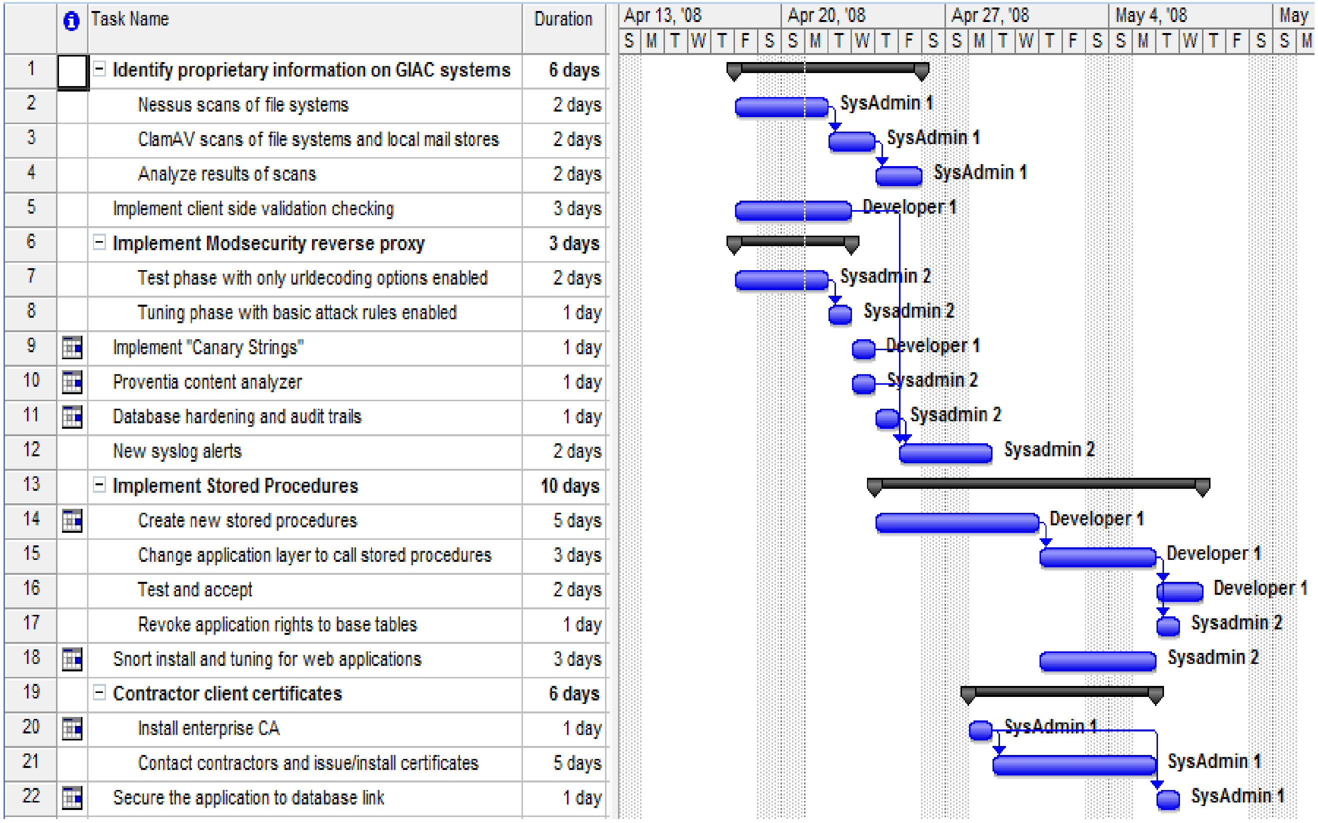
5. Project Management

In order to complete the planned changes in the 15-day timeline, we suggest the following resources:

- 2 System Administrators – The system administrators will be responsible for implementing the planned infrastructure changes including the new reverse proxy, IDS, and enterprise certificate authority.

- 1 Web Application Developer – The developer will be responsible for updating the fortune cookie application to use stored procedures and additional validation checks that can generate alerts when sensitive information is accessed in an unauthorized manner.

Our proposed schedule is below



	ℹ	Task Name	Duration
1	□	**⊟ Identify proprietary information on GIAC systems**	**6 days**
2		Nessus scans of file systems	2 days
3		ClamAV scans of file systems and local mail stores	2 days
4		Analyze results of scans	2 days
5		Implement client side validation checking	3 days
6		**⊟ Implement Modsecurity reverse proxy**	**3 days**
7		Test phase with only urldecoding options enabled	2 days
8		Tuning phase with basic attack rules enabled	1 day
9	▦	Implement "Canary Strings"	1 day
10	▦	Proventia content analyzer	1 day
11	▦	Database hardening and audit trails	1 day
12		New syslog alerts	2 days
13		**⊟ Implement Stored Procedures**	**10 days**
14	▦	Create new stored procedures	5 days
15		Change application layer to call stored procedures	3 days
16		Test and accept	2 days
17		Revoke application rights to base tables	1 day
18	▦	Snort install and tuning for web applications	3 days
19		**⊟ Contractor client certificates**	**6 days**
20	▦	Install enterprise CA	1 day
21		Contact contractors and issue/install certificates	5 days
22	▦	Secure the application to database link	1 day

We recognize that updating our main revenue-generating application is a complex undertaking; however, the changes we are making do not affect the business logic, only the data access layer. With most programming projects, the success and speed of the project will be determined by who is assigned to work on it. If management assigns a developer who is already familiar with the application and working with MySQL, then there is a good chance the project can be completed on time. The creation of the new stored procedures should be closely monitored, and if the developer is not on track, management will need to assign additional developers to meet the deadline. The project schedule is designed to complete the easier, shorter tasks first so that if the application update cannot be completed, there will still be improvements to show the potential buyer.

5.1. Phase III

Not all possible improvements will fit into a 15-day schedule. In the future, we recommend the further tightening of the Modsecurity application firewall from a default-allow stance to a default-deny stance. In the initial implementation, there is not enough time or labor to analyze the application fully and identify its correct behavior. We also recommend encrypting sensitive information as it is stored in the database to protect that information when it leaves via offsite storage of backup tapes.

Further research or custom programming may allow for the searching of sensitive information inside database, mailboxes and other complex formats.

References

1 Matt Hines, *InfoWorld,* "TJX data heist confirmed as largest ever", http://www.infoworld.com/article/07/03/29/HNtjxfiling_1.html, Mar, 2007.

2 Roger Nebel, *Compliance and Governance Digest*, "PCI DSS 3.1 Best Practices", http://searchfinancialsecurity.techtarget.com/tip/0,289483,sid185_gci1299167,00.html, Feb. 2008.

3 Meyer & Ruppert, "STI - Joint Written Project – eDiscovery", http://www.sans.edu/resources/student_projects/200710_002.pdf, Sept. 2007.

4 Ibid.

5 Symantec Corporation, Press Release, http://www.vontu.com/news/releases/592_release.asp, Oct. 2007.

6 Wikipedia, http://en.wikipedia.org/wiki/Canary_trap, Visited April 16th, 2008.

7 IBM, "IBM Proventia Content AnalyzerGuidelines", http://documents.iss.net/literature/proventia/ContentAnalyzerGde.pdf, Dec. 2007.

8 http://www.snort.org/

9 http://www.nessus.org/products/lce/

10 Tenable Security, Nessus Documentation, http://www.nessus.org/plugins/index.php?view=single&id=24760, 2007.

11 Sourcefire, Press Release, http://investor.sourcefire.com/phoenix.zhtml?c=204582&p=irol-newsArticle&ID=1086564&highlight=, Dec. 2007.

12 MySQL 5.0 Reference Manual, Chapter 19: Stored Procedures and Functions, http://dev.mysql.com/doc/refman/5.0/en/stored-procedures.html, n.d.

13 Dr. Eric Cole, Proving Web Vulnerabilities redux, http://www.youtube.com/watch?v=gpZsw7LSrSc, Mar. 2008.