Hash – All Smoke or Stronger is Better?

ISM5700: Situational Response Practicum I

Authors: Andre Shori, ashori<at>mastersprogram.sans.edu
Matt Freeman, mc.freeman808<at>gmail.com
Ronald Tallman, ronald.f.tallman<at>boeing.com

Advisor: Dr. Johannes Ullrich
Accepted: 10 May 2017

1. Executive Summary

GIAC Enterprises is a small to medium-sized supplier of Fortune Cookie sayings. GIAC's Fortune Cookie database is the core intellectual property of the organization and this database is updated via GIAC contractors. Fortune Cookie sayings are submitted via VPN into a web based application server and then finalized in the GIAC Fortune Cookie Database server. As part of this finalization, the resulting file is hashed for versioning and part of the databases integrity controls.

The current hashing algorithm being utilized is SHA1, which is depreciated and of increased risk of hash collisions, possibly resulting in database integrity and versioning issues. This report explores the current vulnerabilities of SHA1 and the accompanying risks and impacts.

As a result of close examination of the current issue; possible mitigations and associated costs, it is the recommendation of this team that the current SHA1 algorithm is retained and appropriate compensatory controls implemented to minimize risk exposure.

2. Current Situation

SHA1 has been largely depreciated[1] (Microsoft, 2016) as a hashing standard due to the increased probability of being able to force a hash collision. A hash collision is a condition where two different messages or files can have the same hash value[2] (Stevens, Bursztein, Karpman, Albertini, & Markov, n.d.). This collision can be accidental in nature or may be the result of a deliberate action. More information on SHA1 and a detailed technical explanation can be found in Appendix A.

SHA1 hash values are utilized in GIAC's Fortune Cookie submission tool and the main Fortune Cookie Database to track file versioning and for database integrity. All information in the Fortune Cookie Database is categorized as Critical/Confidential. SHA1 is not used in any current security controls to either server.

2.1 Upstream Considerations

When Fortune Cookie sayings are submitted by GIAC contractors via secure VPN to the GIAC Fortune Cookie application web server, the resulting file is processed by GIAC Quality Control (QC). Upon approval, the document is submitted to the back end Fortune Cookie database, where it is hashed using SHA1 to enable version control and database integrity. The hash then ensures that there is no pre-existing Fortune Cookie duplicate sayings already in the database.

2.2 Downstream Considerations

Once the resulting file has been confirmed by QC as unique, appropriate and original, a transaction is then submitted to accounts payable in finance for payment processing to GIAC contractors.

---

[1] Microsoft. (2016, November 18). SHA-1 deprecation countdown - Microsoft Edge Dev BlogMicrosoft Edge Dev Blog. Retrieved from https://blogs.windows.com/msedgedev/2016/11/18/countdown-to-sha-1-deprecation/

[2] Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (n.d.). *The first collision for full SHA-1*. Retrieved from https://shattered.io/static/shattered.pdf

A key differentiator for GIAC in the Fortune Cookie market is our ability to send unique Fortune Cookie sayings to Fortune Cookie manufacturers for printing. The hash value helps ensure that there are no duplicates in transmitted print files.

3. Proposed solution

The Fortune Cookie Database is located on our backend, hardened and highly secured server. Access to the data is in line with current GIAC standards and policies regarding critical and confidential proprietary information. Access controls are carefully administered by operations and the data owners with sufficient separation of duties, approvals and monitored logging via the GIAC SOC.

Known, deliberate hash attacks are highly unlikely to succeed. Direct access to the data and the ability to modify or corrupt the database is virtually nil. In addition, sufficient BCP and DRP processes are in place to recover from any incidents involving the server and database.

In GIAC context, SHA1 collisions would only result in possible hash duplications, which is primarily an integrity concern at this point and should be examined in that context. With the implementation of compensating controls proposed in Section Four of this report, the risk to the organization is minimized and any residual risk well within GIAC's risk appetite.

Given that SHA1 is used in our business process purely as version control and for Fortune Cookie database integrity, keeping the existing encryption standard would have the least impact on the organization.

4. Alternate Solutions

4.1 Upgrade to Current Standard

The current replacement standard for SHA1 digital signatures is SHA224, SHA256, SHA384 and SHA512 or as they are more commonly collectively known, SHA2[3] (TBS Internet, n.d.). SHA2 would further reduce the likelihood of hash collisions in the database, given its increased mathematical complexity. SHA2 does not remove the possibility of accidental hash collisions[4] (Keycdn.com, n.d.), although collisions are less probable than SHA1.

Considerable investment is required when upgrading to SHA2. Rework will include enhancement of both upstream and downstream processes and tools, especially the Fortune Cookie Submission application, to support SHA2. Consequently, affected internal and external users will require retraining. The Fortune Cookie database will also need upgrading to support the new standard, with associated implementation risks on our most critical intellectual property. There is also an estimated 2% increase in hashing performance requirements on the Fortune Cookie application server. This increased overhead is within current capacity and of no impact. It is estimated that an upgrade of this nature would cost USD35,500. Detailed costing for this upgrade is located in Appendix B.

[3] TBS Internet. (n.d.). All about SHA1, SHA2 and SHA256 hash algorithms. Retrieved from https://www.tbs-certificates.co.uk/FAQ/en/sha256.html

[4] Keycdn.com. (n.d.). SHA1 vs SHA256. Retrieved from https://www.keycdn.com/support/sha1-vs-sha256/

4.2 Replace the Hash

Alternatively, eliminating the current vulnerability by utilizing a different integrity and version control such as a unique index key field in the database would avoid current risk exposure. This solution entails the same projected costs to rework the database and toolsets as the upgrade of the current hash standard, minus the performance requirements. It is estimated that this rework will cost USD25,500. Detailed costing for this upgrade is located in Appendix B.

5. Compensating Controls

The risk and impact of possible hash collisions that are inherent in all hashing standards must be addressed. Given the criticality of the GIAC Fortune Cooking database and the associated submission processes previously outlined, retaining SHA1 as the current standard will require the implementation of additional immediate and long-term compensating controls.

Duplicate hashes in the Fortune Cookie database are unacceptable, given the need for proper version control. An automated process to search for any existing and future duplicate hashes must be added to current database management. Should duplicates exist, the database administrator, utilizing existing change management, would only need to open the file and make any small change such as the addition of a space or hidden character. By saving the altered file, the hash is regenerated, resulting in a new, unique value.

It is further possible that users may be impacted by hash collisions, although a search on GIAC Help Desk records yielded no such previous incidents. GIAC downstream customers may someday encounter duplicate hashes in one of the Fortune Cookie print files sent to them. GIAC Quality Control must add a check before sending print data to ensure that no duplicate hashes exist. This can be automated and should result in minimal additional overhead. Coupled with the additional database management controls, this will eliminate any possibility of print file corruption.

GIAC Help Desk technicians will need to be informed and trained on new solutions for any reported issues related to a possible hash collision. If a hash already exists in the Fortune Cookie database, upstream contractors submitting Fortune Cookie sayings may encounter a “duplicate submission” error. GIAC Help Desk will need to instruct affected contractors to make a change such as adding a space to the submission to generate a different hash. The workaround should be included in the GIAC Help Desk knowledge base, and incidents of this nature should be tracked as part of the Help Desk daily report. A sharp increase in calls of this type may point to a problem with the Fortune Cookie submission tool or potential Fortune Cookie database integrity issues.

Longer term compensatory controls include adding hash collisions to the GIAC risk register and including it as a known vulnerability in GIAC’s annual risk assessment exercises. A request to the GIAC IT steering committee to add this as a possible consideration to GIAC BCP/DRP scenarios will ensure that any resulting disruption is recoverable within GIAC RTO and RPO thresholds.

As technology improves or there are additional increases in GIAC contractor Fortune Cookie submissions over time, the probability of an incident may grow. This risk should be added to GIAC annual audits and included in future vulnerability scans on our database and application servers. SHA1

vulnerabilities are also a useful condition to add to subsequent GIAC internal and external penetration testing to ensure that new or unknown exploits are unable to abuse SHA1 encryption nefariously.

Implementation of the above compensating control costs is estimated at USD1,800. A cost breakdown is available in Appendix C.

6. Conclusion

SHA1 is a depreciated hashing standard containing known vulnerabilities that could potentially lead to version control issues and Fortune Cookie database integrity issues. Within the context of GIAC's current implementation of SHA1, this risk is deemed to be minor. Together with current and recommended compensatory controls, risk and impact are reduced to levels within GIAC's risk appetite for critical/confidential data.

Budget for compensatory implementation is estimated at USD1,800. With Executive Management sponsorship, this project can be added to GIAC's current project portfolio this quarter.

Appendix A – SHA-1 Technical Exploration

The Secure Hash Algorithm (SHA) is a type of cryptographic hash function that ensures data has not been modified. SHA accomplishes this by computing a cryptographic hash value for a given piece of data that is unique to that specific input. The United States National Institute of Standards and Technology developed the SHA family of hashing algorithms, which includes the SHA-1 hashing algorithm. [5]

A hash function is designed to yield a unique hash value (digest) for every piece of distinctive data as input. Therefore, differing hash values are key to determining if data has been compromised or altered. As a cryptographic requirement for widespread use, finding two messages (inputs) that lead to the same digest should be computationally infeasible. [6]

Because hash functions have infinite input length and a predefined output length, there is inevitably going to be the possibility of two different inputs that produce the same input hash. When two separate inputs produce the same hash output (digest) it is defined as a collision.

Collision resistance is a property of cryptographic hash functions. For example, a hash function is resistant if it is hard to find two inputs that hash to the same output. That is two inputs a and b such that H(a) = H(b), and a is not equal to b. [7]

Cryptographic functions are usually designed to be collision resistant. But many hash functions that were once thought to be collision resistant were later broken.

After more than 20 years after its introduction a practical technique has been published that can generate collisions using SHA-1. This represents the culmination of 2 years of research between the CWI institute and Google. Google has advocated for years the deprecation of SHA-1 most notably for the signing of TLS certificates.[8]

[5] Understanding SHA-1 Vulnerabilities — Is SSL No Longer Secure? - Entrust, Inc. (n.d.). Retrieved from https://www.entrust.com/understanding-sha-1-vulnerabilities-ssl-longer-secure/

[6] Google Online Security Blog: Announcing the first SHA1 collision. (n.d.). Retrieved from https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

[7] Collision resistance - Wikipedia. (n.d.). Retrieved May 10, 2017, from https://en.wikipedia.org/wiki/Collision_resistance#cite_note-GoldwasserBellare-1

[8] Google Online Security Blog: Announcing the first SHA1 collision. (n.d.). Retrieved from https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

Appendix B – Detailed Costing for Alternative Solutions

Option 1: Upgrade to modern hashing standards	Estimated hours	Estimated Costs
Retool upstream customers	40	$6,000.00
Retool downstream customers	40	$6,000.00
Rewrite web sumisssion app	80	$12,000.00
Process training	10	$1,500.00
Additional process overhead		$10,000.00
Total Hours	170	$35,500.00
Option 2: Change to proper unique index key		
	Estimated hours	Estimated Costs
Retool upstream customers	40	$6,000.00
Retool downstream customers	40	$6,000.00
Rewrite web submission application	80	$12,000.00
Process training	10	$1,500.00
Total Hours	170	
Total Cost USD		$25,500.00

Appendix C – Detailed Costing for Compensatory Controls

Compensatory Control Estimates		
Optimize and add to DB integrity checks	3 man hours x $100	$300
DB manual corrections	5 man hours	$500
2 hours help desk training x 5 Representatives	10 man hours X $100	$1,000
Estimated Total Cost		$1,800.00

Hash – All Smoke or Stronger is Better?

Andre Shori Ronald Tallman Matt Freeman

11 May 2017

ISM5700 Situational Response Practicum
SANS West 2017

Objective

- Review Potential Threat to Fortune Data
- Assessment of Current Impact to Operations
- Review Mitigation Scenarios

SHA-1 Vulnerability Summary

- SHA-1 Algorithm No Longer ‘Safe’ From Forced Hash Conflict
- Attacker Could Compromise Fortune Data
  - Attacker Would Require Original Hash and Access to Servers to Inject Compromised Data
- Unintentional Hash Conflict More Likely

GIAC SHA-1 Usage

- GIAC uses SHA-1 to Ensure Integrity and Version Control
- Fortunes Submitted Through Web Application
- Validates Fortune Writer Payments
- Quality Control For Printing Partners

Compensating Controls

- Automated Detect and Correct
- Topic Specific Training
- Monitoring and Reporting

Possible Courses of Action

- Continue Using SHA-1
  - Implement Compensating Controls and Monitor
- Hash Modernization Project
- New Version Control Solution

Conclusion

- Recommend Continued Use of SHA-1 with Compensating Controls
- Feedback From Executive Team
- Next Steps

Thank You

Questions & Answer

400 – 410 reviewed the assignment and scope of task
410 -425 network connectivity and voip troubleshooting for group
430 -440 began initial research of SHA-1 concern
442 discussed roles and responsibilities. Matt to CIO report, build PP, and tentative presenter
445 started researching!
Understand what the potential attack is
https://learncryptography.com/hash-functions/hash-collision-attack
A **Collision Attack** is an attempt to find two input strings of a hash function that produce the same hash result. Because hash functions have infinite input length and a predefined output length, there is inevitably going to be the possibility of two different inputs that produce the same output hash. If two separate inputs produce the same hash output, it is called a **collision**. This collision can then be exploited by any application that compares two hashes together – such as password hashes, file integrity checks, etc.
Practically speaking, there are several ways a hash collision could be exploited. if the attacker was offering a file download and showed the hash to prove the file's integrity, he could switch out the file download for a different file that had the same hash, and the person downloading it would be unable to know the difference. The file would appear valid as it has the same hash as the supposed real file
look at some experts opinions
https://www.schneier.com/blog/archives/2017/02/sha-1_collision.html
how is it being reported in the news
https://www.theregister.co.uk/2017/02/23/google_first_sha1_collision/
500 matt's back of the envelope plan v1:
- IT team needs to understand how a sha-1 conflict can impact our business
Assign team to research sha-1, including how the attack can be used. Is it currently being used in attacks, how and what impact
- Evaluate our internal IT environment for potential vulnerabilities to sha1 conflict
- Create an initial assessment of the impact of a successful sha1 attack
- Prepare mitigations for possible attacks
- Assess costs and process of moving to more complex algorithm
- Provide recommendations to CIO:
o Do nothing, vulnerable but low probability
o Begin low urgency upgrade and migration plan while deploying compensating controls
o Immediate mitigation and migration
Included costs:
Recoding
Additional computational resources
training?

514 Did some math, 6000 computational hours is 250 days
525 sha cracker's research paper https://shattered.io/static/shattered.pdf
https://arstechnica.com/security/2017/02/at-deaths-door-for-years-widely-used-sha1-function-is-now-dead/
In the meantime, the researchers have released a tool that detects if files are part of a collision attack. Had the researchers performed their attack on Amazon's Web Services platform, it would have cost $560,000 at normal pricing. Had the researchers been patient and waited to run their attack during off-peak hours, the same collision would have cost $110,000.

534

Started a problem statement:
The current primary product of GIAC Enterprises is the content of the fortunes themselves, the data. Data is stored and processed in 2 data centers at highly rated colocation facilities, one in the U.S. and one in Indonesia.
we don't use SHA-1 for security, only for version control of our fortunes.
Knowns:
SHA-1 is currently used for version control of sensitive data.
Research has been released that shows attackers could force collision for hashing algorithms for SHA-1
Unknowns:
Other security controls in place for protection fortune cookie data
Other uses of sha-1 in internal network
Mitigations:
Asses internal vulnerability
Plan migration
Use tool to validate
555 started powerpoint draft

=======
RT log notes
Log Summary
Team defined roles for implementing assignment

Team Meeting #1 notes/assignments 0730hrs:
• Assignment
• Project plan draft
• Contact details
• Roles and responsibilities
o Project plan – Andre
RACI chart -
Timeline –
WBS –
Risk register (project) -
o CIO report - Matt
Assumptions
Assumption checking
o Research - All
o Presentation draft - Matt
o Presentation live – Matt?
o Uploading – Andre
o Lab notebook – Ron
• Send lab notebook to group

Problem Statement (?)
GIAC uses SHA-1 as version control only
5:30PM pacific
Questions that are coming to the forefront:
Are we using SHA-1 in any other part of the business? Are we at risk elsewhere?
What are the processes in publishing the fortunes? How are they protected?
What is GIAC risk tolerance?
Impact if our system is compromised?

VPN encryption and hashing standards?
How easy and costly are the mitigation measures?
5:45 - (research)
What are the internal measures in place that protects the data/fortunes?
https://techcrunch.com/2017/02/23/security-researchers-announce-first-practical-sha-1-collision-attack/

5:50 - (research)
What is cost to generate a collision?
https://www.appsecconsulting.com/blog/practical-advice-for-sha-1
6:00 - (research)
What is the real risk?
Linus Torvalds says the sky isn't falling
https://news.slashdot.org/story/17/02/25/2229253/linus-torvalds-on-gits-use-of-sha-1-the-sky-isnt-falling
6:15 - (research)
Regroup for assessment
Matt working thru executive summary draft

6:30
Group assessed current tasks – agreed to continue tasks up to 8:00pm in the room.
Group agreed to close down collaboration (remote/hotel) at midnight and resume at 8:00 am.

6:40
Matt sent email to schedule time with Dr. Ulrich for technical questions

6:59 pretty reasonable approach to assess and mitigate (for root certs but process seems worth digging into)
https://blog.qualys.com/ssllabs/2014/09/09/sha1-deprecation-what-you-need-to-know

12:31 SG time - (on Skype to the group) Some questions I think need to be answered on this assignment:
• What is the CIO's real concern here?
• Where is the hash generated? Client side? Server side?
• What's the impact of a hash collision?
• What controls and risk mitigations are available to counter this?
• What are we proposing? Why?
• What's the additional cost of our solution?
• Perhaps we can accept the risk for now and monitor some factors we determine to assess this as part of our annual risk assessment?
• Include this as part of our vuln and pen testing?
• What other controls do we have already in place on fortune submissions? VPN and what else?

From the research I see on Matt's notebook, it appears that using SHA-1 may be a low risk to the organization.
Do you concur?
12:37 Seems to me that a hash collision wouldn't benefit the contractors, since they need each submission to be unique to get paid.
12:38 is this more of an integrity issue on our database, that perhaps we might generate a hash collision inadvertently and two diff files appear as duplicates?
and if so, is the 2% additional computation something that is actually negligible? Could we upgrade to SHA-256 for eg and use the same hardware? For example, if we just recompute (upgrade) the hash only when we save a file? Or when we run our nightly/monthly batch billing which would open the file, check the contents and if valid, pay the submitter? If we assume say 60,000 fortunes per hour (the files really cant be all that big) that's about 20 extra computations per minute or 1,200 less completed billing processes per hour. I don't see us getting that many fortunes per day (I did send Dr Ullrich this question) so would this really mean that our biggest process, which I assume to be billing, would only add 1.2 minutes processing time?
12:44 Someone check my math please :p

12:48 SG time - **browser compatibility with SHA256 (SHA2)** - all major browsers support it including mobile browsers
https://support.globalsign.com/customer/portal/articles/1499561-sha-256-compatibility#1a

12:57 SG time - How many fortunes are generated per day? Not many… **World's largest company only has a DB of 15,000 sayings IN TOTAL.** They update their DB every few years.
http://mentalfloss.com/article/50610/who-writes-messages-fortune-cookies

Seems that one assumption we can challenge is the # of submissions, might be a low #! Impact may be minimal to the business in terms of increased overhead (processing time).

Perhaps there are internal processes that generate higher overhead such as opening the file to use it to make a cookie. Then we're looking at about 5 million cookies per day (based on the previous link) but I'm not certain that each cookie would have a unique fortune printed on it (perhaps it's a differentiator from our competition?). If we assume that each fortune is unique, and we use a reasonable estimate of 625,000 unique fortunes/hour (I'm using an 8 hour working day for this calc) sent to the printer (again assuming we have some sort of high speed printer capable of handling this) and hashes play a part in this process by ensuring we don't print duplicates, then a 2% decrease in performance equates to an increased total daily processing time of ~1.6 minutes? (2% of 8 is 0.16).

Let's look at this from a different perspective - **what is the CIO's real concern here?**
Is it increased processing cost because we have to switch hashes? Probably
Is it increased risk of fraud by contractors using SHA1 collisions to their advantage somehow? Not sure this pans out, can't think of how that would benefit them so unlikely.

Is it increased risk of paying out more than what we should? Or alternatively, paying out less than what is due to contractors because we see a duplicate instead of two unique files? Possibly legal ramifications and costly manual checking.

Is the CIO just not understanding what the net impact of SHA1 collisions are on our business? Could there be no impact at all? Can we prove this?

Collisions are inevitable regardless of the hashing algorithm used. It's unlikely but not impossible.

Collision resistance is a property of cryptographic hash functions: a hash function *H* is collision resistant if it is hard to find two inputs that hash to the same output; that is, two inputs *a* and *b* such that $H(a) = H(b)$, and $a \neq b$.[1]:136

Every hash function with more inputs than outputs will necessarily have collisions.[1]:136Consider a hash function such as SHA-256 that produces 256 bits of output from an arbitrarily large input. Since it must generate one of 2^{256} outputs for each member of a much larger set of inputs, the pigeonhole principle guarantees that some inputs will hash to the same output. **Collision resistance does not mean that no collisions exist; simply that they are hard to find**.[1]:143

The "birthday paradox" places an upper bound on collision resistance: if a hash function produces *N* bits of output, an attacker who computes only $2^{N/2}$ (or {\displaystyle \scriptstyle {\sqrt {2^{N}}}}) hash operations on random input is likely to find two matching outputs. If there is an easier method than this brute-force attack, it is typically considered a flaw in the hash function.[2]

Cryptographic hash functions are usually designed to be collision resistant. But many hash functions that were once thought to be collision resistant were later broken. MD5 and SHA-1 in particular both have published techniques more efficient than brute force for finding collisions.[3][4] However, some hash functions have a proof that finding collisions is at least as difficult as some hard mathematical problem (such as integer factorization or discrete logarithm). Those functions are called provably secure.[2]

https://en.wikipedia.org/wiki/Collision_resistance

13:23 SG time - Perhaps this is not a concern at all, since all hashing algorithms suffer from this weakness and compensating controls such as a manual check when there's a repeated hash value + low probability make this unlikely. Forcing a collision wouldn't really do anything useful as far as I can tell.

So perhaps the CIO's concern here is more of data diddling and security. Since we're not using hashing as part of security, only version control (db integrity) and since it's likely from the assignment brief "we don't use SHA-1 for security, only for version control of our fortunes" - Karen, then the CIO needs to be reassured that this is not a real issue. SHA1 for version control on the DB is fine, we're not using it for message digest functions during fortune submissions.

13:30 SG time - Looking at the alternatives/upgrades now, increasing to SHA256 (SHA2) would increase printing time potentially by about 1.2 minutes (if we even print the cookies, not just supply the fortune cookie sayings. Maybe our customers also use the hashes when printing?) If we need to support that we don't actually print the cookies, we can refer to my 30 group project, where this is stated.

Other upgrade risks

The increase in processing time is minimal, however there is increased risk to the database while we have two different hashing algorithms at play.

There is potentially a need to redevelop our fortune cookie submission application to allow/generate SHA2. We probably need to upgrade the main database, which could be an expensive, careful project because this is critical, top secret data. We would have to duplicate this to both processing centres (colocated but are they actually mirrored? What's the RTO, RPO? Never mind, that's a diff issue). So let's assume that they are mirrored, so we'd need to rollout at both sites using good change management.

We might need new hardware but that's not likely since it's a 2% overhead increase. We should have the overhead to support this, otherwise I'd fire whoever is doing capacity management.

Our customers may need to retool as well, if they use the hashes in their processes (printing example from before).

Compensating controls

We could implement controls to look for duplicate values in the database, and manually open and make one small change to the file (add a space for eg) to force it to recompute the hash.

We could even create a script to do this, however better to have an approvals process in place for SoD and logging.

We can respond to any complaints from our customers about duplicate hashes by informing our help desk of the possibility (knowledge management, help desk DB) and implement procedures to handle these situations (which should be very rare). We can even track and alert on if there's an uptick on these types of calls and send to the SOC (could be actual fraud/attack/indication of other issues in the DB) and trigger an investigation.

14:06 SG time - Illustrate to CIO the probability of unintentional SHA1 collision:
Worst case (to date) sha1 collision probability
The SHAppening

On 8 October 2015, Marc Stevens, Pierre Karpman, and Thomas Peyrin published a freestart collision attack on SHA-1's compression function that requires only 2^{57} SHA-1 evaluations.

https://en.wikipedia.org/wiki/SHA-1

Or about 1 in 144,115,188,075,855,872.

In contrast, the odds of winning the powerball lottery are 1 in 292,000,000 (thats 9 less digits).

http://graphics.wsj.com/lottery-odds/

19:59 SG time- reply from Dr Ullrich

ISE\ISM 5700 Technical Advisory Requests

Ullrich, Johannes <redacted> 10 May 2017 at 19:26

To: <redacted>

20:00 SG time - 2nd email reply from Dr Ullrich arrives

Ullrich, Johannes <redacted> 10 May 2017 at 19:26
To: "<redacted>

> General comment: If you aren't sure about any details of the assignment: Make them up. You are assumed to be working at GIAC Enterprises for a while, so you are familiar with the network/business and what you assume, as long as it is reasonable, is true. In the scenario, I am considering "Fortune Cookie" always as a placeholder for "confidential information".
>
> What is the current approximate cost per fortune cookie upload to GAIC? What are we paying the contractors per fortune uploaded?

$500 per fortune and about 10 per day (responding to your second email)

- Are the contractors using a VPN+web based submission tool? Or are they using a custom client-side application?

VPN + Web (or other standard tools, no custom client)

- Can we assume the fortune cookie submissions are categorized as business important/confidential information? Or is it business critical/confidential?

The critical part is the integrity of the fortunes. Fortune confidentiality is only critical for new fortunes that have not been released yet.

- Does the fortune cookie submission mechanism pose any risk at all to the main fortune cookie database or can we assume that the main DB is on a separate hardened back end server and at no known risk at this time?

Up to you to decide. But I think you can assume that once a fortune when through an “intake” process, it is stored in a read only database. But the integrity of this database still needs to be verified.

- Is it acceptable to assume that the fortunes submitted are checked via an existing process for uniqueness (not a repeat of a previous fortune) and then approved automatically for payment (finance/accounts payable) at month end? This ties in with non-repudiation (version control) of the fortune submission, which is what we are assuming the hashing function is for.

The hashing is for uniqueness but also to check that the fortune dos not get altered.

- Are fraudulent transactions where a contractor submits invalid fortunes with valid pre-used hashes an invalid concern? Can we safely assume that there is a control mechanism in place to detect multiple hashes with the same values? Do we need to state this to the board? How technical should we assume the BOD are? May we assume both you and Chris are technical experts when you're acting as the BOD tomorrow evening?

As far as invalid submissions go: Also consider a fortune write working on a compromised system. But they wouldn’t be able to submit the same fortune multiple times.

You should assume some familiarity with the network and technical aspects of how existing processes work. But do not assume the audience understands the difference between SHA1 and SHA256 (nor has a lot of interest in the details). So it isn’t “Chris the SANS Instructor” you are talking to, but “Chris the CEO” who has a business degree and some working experience/familiarity with GIAC IT systems.

Looping in Krysta for her info.

<redacted>

ISE\ISM 5700 Technical Advisory Requests

Ullrich, Johannes 10 May 2017 at 19:29
To: MC Freeman <redacted>

· Do we know the risk appetite of the business?

See my earlier email, but the critical issue is integrity of the fortunes (so they do not get modified after submission)

· Any insight into how they are securing vpn of remote users?

"Best practices". So assume some form of PKI

· What are the processes used to publish the fortunes from end users?

If an end user is a fortune writer: The fortune writer connects to the VPN and submits the fortune via a custom intranet web app

.

· Are there other internal controls already in place that is protecting the fortune data?

Yes. Think about standard system hardening and backups.

Thanks for any guidance you can offer!

Matt, Andre, Ron

1900 SG time - Rewrote the outline to be more business focused and added details. Uploaded to google docs in prep for team meeting @ 0800 US time.

07:50 Pacific
Dr. Ullrich's response does support our team establishing some reasonable assumptions that GIAC Enterprise is currently employing for their security posture. I think it reasonable that we can assume best practice VPN access and internal FW controls are in place that safeguard access to the intellectual property that GIAC owns and controls.

Team meeting scheduled @ 0800 US time
Regroup and strategize the remaining 8 hours.

Regroup at 0830
Agreed on solution setup - Andre built framework/outline for CIO report - now tasked to create report for group.
Ron building Appendix for SHA-1 problem definition
Matt building PowerPoint for presentation

04:13 SG time - CIO report final draft completed. Sent to team for final review.

04:28 SG time – CIO report, lab notebook finalized and approved by team for submission. Submitting now, this is the final update.

ISM5700 Assignment – Hash - all smoke or is stronger, better?
Project Plan

Contents

Assignment Topic:
Hash - all smoke or is stronger, better?

Assignment Scenario:
Your company, GIAC Enterprises, is a small to medium-sized growing business. It employs 1,500 employees, including 750 business and IT workers at corporate headquarters, 250 employees at the Indonesian office, and the remainder remote workers distributed worldwide. The servers are more diverse; almost all of them run Linux. The company is the largest supplier of fortune cookie sayings in the world. It prides itself on a rich history, as well as cutting edge original research. The current primary product of GIAC Enterprises is the content of the fortunes themselves, the data. Data is stored and processed in 2 data centers at highly rated colocation facilities, one in the U.S. and one in Indonesia.

GIAC CIO/CISO, Karen Brown, walked into the office of one of the senior engineers, Chris Green, and noticed a news story on his screen from the GIAC Advisory Board.

"What's up?" asked Karen.

Chris said, "Well I may be a Nervous Nellie, but I have been reading about the forced collision on SHA-1."

"Okay," Karen said, "but we don't use SHA-1 for security, only for version control of our fortunes. And it was about 6,000 computation hours to force a collision. To switch to SHA-256 would add something like 2% computation per fortune, after the recoding, at least that is what my IT guys are telling me."

"We count on our fortune writers to give us quality fortunes. They submit them via a VPN. We only use a hash to track the version, but if you think we are at risk, I will put a team on it."

Assignment Charter:
Propose a solution while assessing the cost, risk, and impact of potentially migrating from SHA-1 encryption to a better, more modern encryption standard.

Assignment Stakeholders:
- Andre Shori
  (project team member)
  <redacted>

- Matt Freeman
  (project team member)
  <redacted>

- Ronald Tallman
  (project team member)
  <redacted>

- Chris Crowley
  (GIAC CEO)
  <redacted>

- Dr. Johannes Ullrich
  (Technical Advisor/ GIAC CIO)
  <redacted>

- Krysta Kurzynski
  (Enrollment Manager)
  <redacted>

Scope Management:

Assignment Requirements

- Assess the current encryption standard and propose how to move forward

Scope (assignment)

- Asses the current encryption standard. Understand the current risks and impact to the business.
- Assess the costs, risks and impact/benefits of an improved encryption standard.
- Propose how to move forward in a CIO report and present findings to the Board.

Resources (assignment)

- See Appendix A - RACI Chart

Time Management:

Activities (assignment)

Sequence

- Project Plan – 10% of grade
- CIO Report – 50% of grade
- Presentation – 40% of grade
- Final Project Upload
- Reflections (individual)

Resources

- Project Team - Andre Shori, Matt Freeman, Ronald Tallman
- STI advisors – Krysta Kurzynski
- Technical Advisor – Dr. Johannes Ullrich

Durations

- Project Plan - 2 hours. Deadline – 10 May 2017 1600hrs UTC-8
  - Project plan update - @ 12-hour mark
- Written Report - 24 hours. Deadline – 11 May 2017 1600hrs UTC-8
- Lab Notebook – 24 hours. Deadline – 11 May 2017 1600hrs UTC-8
- Presentation – 1 Hour. Deadline – 11 May 2017 1700hrs UTC-8
- Reflections (personal) - 30 Days. Deadline 10 June 2017 0000hrs UTC -7
- Project Upload – 30 Days. Deadline 10 June 2017 0000hrs UTC -7

Schedule (assignment)

- See Appendix B - Timeline

Cost Management:

Costs/Budget

- There are no costs or budgetary requirements for the completion of this project.
- Any incidental costs will be borne out of pocket by the project team members

Quality Management:

Project Plan (10% of grade)

- This document is the project plan and describes:
  - who is going to do what part of the work
  - how long are tasks expected to take
  - what is the schedule
  - The work breakdown structure

Final Report (50% of grade)

- CIO level executive summary
- 1.5 – 2 pages recommended (5 max)
- Single spaced, double spaced between paragraphs

Presentation (40% of grade)

- Will be 15 minutes long (excluding Q&A)
- 7 Slides – title, objective, four body/content, and conclusion
- Delivered by one nominated team member

Lab Notebook (0% of grade, required)

- Show our research, internet links, interviews, lab tests, etc.
- To be appended to the end of the Final Report

Reflections (0% of grade, required)

- Individual Team Members reflections 1-2 pages long
- Each team member will upload their reflections to Canvas

Upload of Final Project (0% of grade, required)

- Nominated team member to upload the final plan, written report and presentation to Canvas

Quality Metrics

- Quality metrics will be largely assessed from feedback received from Project Advisors and integrated on an ongoing basis by project team members.

Change Requests

- Change requests to the assignment scope will be submitted to the Lead Course Faculty, cc'd to the DPS and STI advisor and only implemented with written approval.

Human Resource Management:

- In the event of disputes or poor deliverables, mitigation actions are described in the risk management section of this project plan.

Communications Management:

Communications Management Plan (assignment)

Stakeholder	Document	Format	Contact Person	Due
Assignment Grading	Project Plan Lab Notebook CIO Report Presentation Reflections	Word Word Word PPT Word	Chris Crowley (CEO), Dr. Johannes Ullrich (CIO)	**11 May 2017**
Project Advisors	CIO Report Presentation	Word PPT	Dr. Johannes Ullrich	**11 May 2017**
STI Advisor	Project Plan Lab Notebook CIO Report Presentation Reflections	Word Word Word PPT Word	Krysta Kurzynski	**11 May 2017/ 10 June 2017**

Risk Management:

Risks

- Time Management
- Time Zones
- Misunderstanding of assignment scope
- Misunderstanding the assignment deliverables
- Misunderstanding on deliverable ownership
- Missing deadlines
- Poor commitment by team member
- Poor deliverables by team member
- Disagreement regarding best approach to the assignment

Risk responses

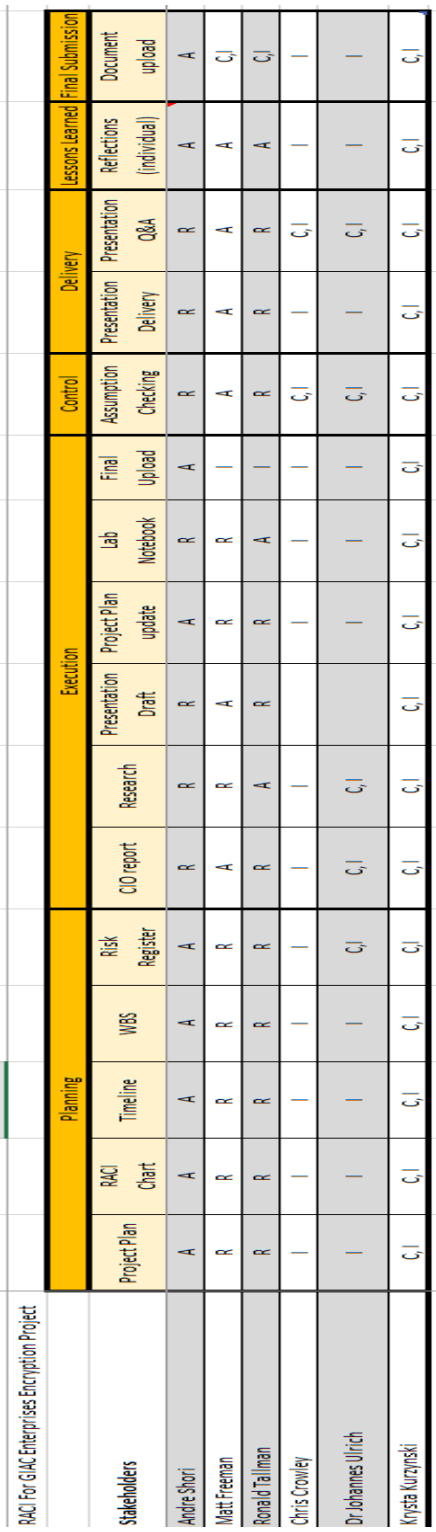
- Time Management – team members have agreed to be available for contact with each other via Skype at all times.
- Time Zones – Matt and Ron are located onsite in San Diego. Andre is in Singapore however given the nature of the project and short timeline; this is not expected to present a challenge.

- Misunderstanding of assignment scope – constant checking and requests for feedback from project advisor will ensure clear and definite understanding of the assignment scope.
- Misunderstanding the assignment deliverables - – constant checking and requests for feedback from project advisor will ensure clear and definite understanding of the assignment scope. Clear and constant communication between project team members will ensure that both team members clearly understand the deliverables.
- Misunderstanding on deliverable ownership – utilizing a RACI chart to document deliverable ownership and a clear project timeline will ensure that team members have ownership of each task and milestone are clearly defined.
- Missing deadlines- team members will utilize Skype as well as the project timeline and documentation to ensure that all deadlines are met. Open communication channels between team members will also aid in this effort. All project deadlines are at midnight of the due date (0000hrs of the following day).
- Poor commitment by team member – Open communication, documented deliverables and recorded submissions by each team member will help to ensure that team members are delivering quality and meeting all deadlines. In the event of a dispute, a resolution will be requested by the offended team member from our STI Advisor.
- Poor deliverables by team member – Open communication, documented deliverables and recorded submissions by each team member will help to ensure that both team members are delivering quality and meeting all deadlines. In the event of a dispute, a resolution will be requested by the offended team member(s) our STI advisor.
- Disagreement regarding the best approach to the assignment – In the event of a dispute that cannot be settled between team members on the best approach to the assignment, the team members will consult with our STI advisor, for resolution. The decision by our advisor will be considered final.

Stakeholder Management (assignment):

Stakeholder (name/role)	Importance (High/Med/ Low)	Current Support Level	Desired Support Level	What's important to Stakeholder	What do we need from Stakeholder	Strategy to enhance support
Chris Crowley/ Grading	Low	Low	Low	Project Deliverables Quality of deliverables Completeness	Grading	Presentation on 11 May
Johannes Ullrich Project Technical Advisor/ Grading	High	High	High	Quality of deliverables Completeness	Technical advisory Ideas Guidance Items that we may have missed	Introduce ourselves to Johannes via email. Follow up with an F2F meeting @ the conference if possible to interview him. Follow up with a phone call if necessary. Keep asking relevant questions as they emerge.
Krysta Kurzynski STI Advisor	High	High	High	Project Deliverables Project Final Grade Students GPA Students Engagement Level	Connections to resources Project Submission	Keep updated, status report at each milestone. CC submission of all assignment deliverables to STI advisor.

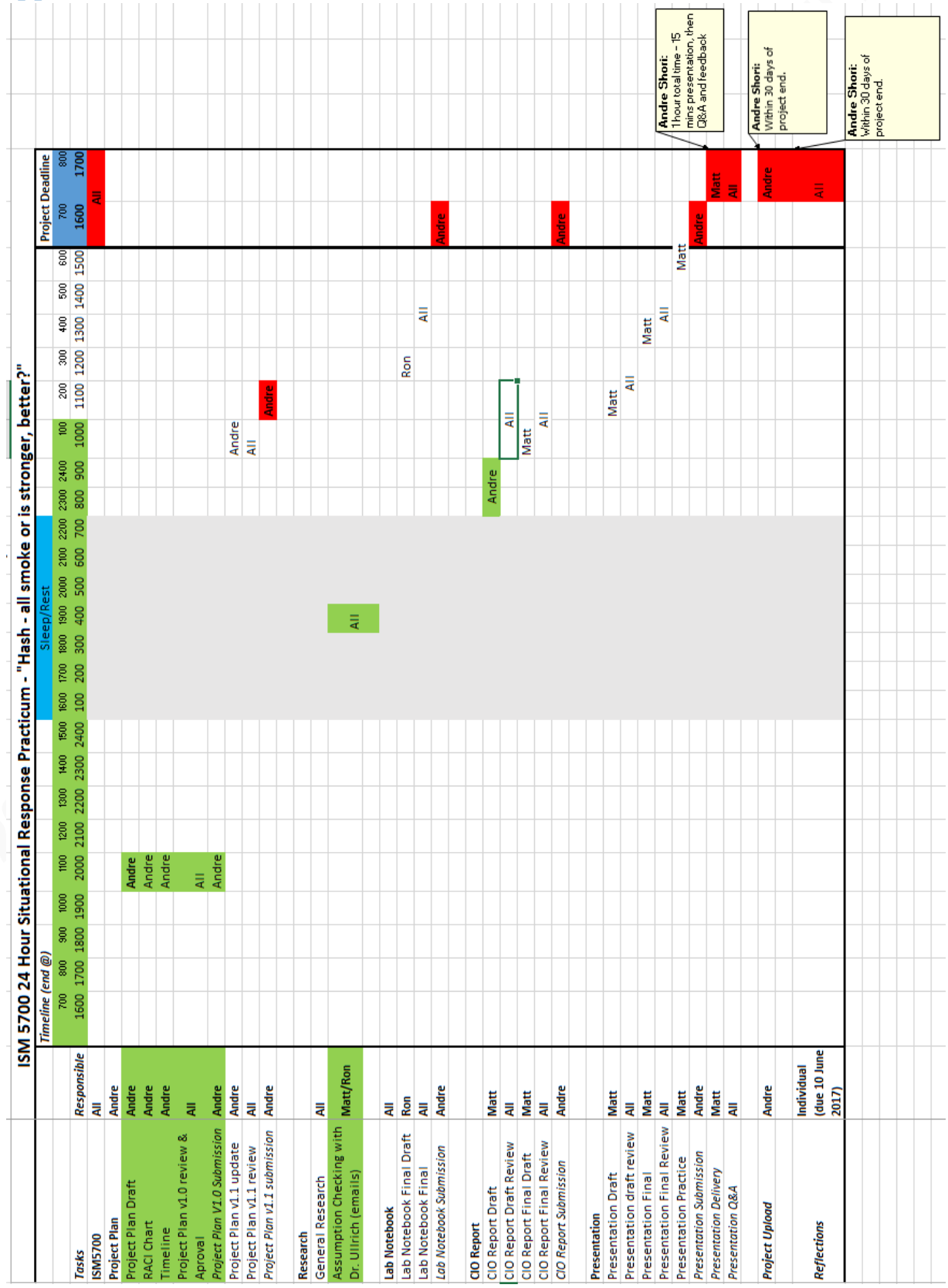
Appendix A -RACI Chart



RACI For GIAC Enterprises Encryption Project

Stakeholders	Planning					Execution						Control	Delivery		Lessons Learned	Final Submission
	Project Plan	RACI Chart	Timeline	WBS	Risk Register	C/O report	Research	Presentation Draft	Project Plan update	Lab Notebook	Final Upload	Assumption Checking	Presentation Delivery	Presentation Q&A	Reflections (individual)	Document upload
Andre Shori	A	A	A	A	A	R	R	R	A	R	A	R	R	R	A	A
Matt Freeman	R	R	R	R	R	A	R	A	R	R	I	A	A	A	A	C,I
Ronald Tallman	R	R	R	R	R	R	A	R	R	A	I	R	R	R	A	C,I
Chris Crowley	I	I	I	I	I	I	I		I	I	I	C,I	I	C,I	I	I
Dr Johannes Ulrich	I	I	I	I	C,I	C,I	C,I		I	I	I	C,I	I	C,I	I	I
Krysta Kurzynski	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I	C,I

Appendix B - Timeline



Andre Shori, Matt Freeman, Ronald Tallman