A Good Fort Has a Gap

Arto Juhola
VTT Technical Research Centre of Finland

17. December 2009

Figure 1: Old painting of Himeji castle [24][21]

1 Name

"A good Fort Has a Gap".

2 Problem

You are not (and can't be) absolutely certain that the principal defences (of a system or a structure) you have set up are impenetrable.

In other words, you cannot out rule the possibility of a successful break-in, since the attacker might be able to circumvent or confuse the provided access control points by some clever means. This might be due to the breadth and complexity of the system or structure, or to some other inherent weakness.

3 Context

You have to defend a site with a well defined perimeter, be that a physical fort, a network or something else.

You have taken care of the more straightforward means of defence. But, confidence in the infallibility of the principal defence might be misplaced, rendering your system or structure to be just a little better off than a sitting duck; You have to cover all possible vulnerabilities, while the attacker needs only a single one.

4 Forces.

1. The site cannot parallel the kind of imperviousness a European, pre-cannon era (1100-1300) castle had against the attackers of the period, when the castle defenders usually could most often just simply wait until the attackers lost heart and returned home[22].

2. You lack applicable methods, resources, manpower, funds and/or time to realise an impenetrable protection, or it is a sheer impossibility.

3. However, you have authority and means, within reasonable bounds, to introduce improvements.

4. Deploying improved (but not perfect) protection prompts intruders to device and use more intricate forms of attack, including ones not foreseen and subsequently lacking means of detection.

5. To make things worse, potential malefactors have often many advantages over you that make foreseeing attacks difficult: Covert collection of preparatory information, gathering of “mass”, concealed identity, concentration of forces, surprise, freedom of means, particularly with regard to law, latitude in manoeuvres and timing, and possibilities to cover up their tracks.

6. Because of the above disadvantages, it might be necessary to diminish the extent of the “fort’s” perimeter”

7. The manpower you have is capable, disciplined and ample.

8. Awareness of any stealthy, on-going break-in attempt is essential in the utmost, since if continuing unnoticed it might eventually lead to the compromising of the “fort”. Also, the sooner the attempt will be noticed the sooner the actions to neutralise it can be started, and consequently, the lesser the damages will be.

5 Solution

Your system or structure should have an intentional weakness (real or fake), visible to the outside. This way you can lure attackers to enter in a way you can predict and detect.

Behind the “gap”, you are able to select and set the “stage” to your liking, watch it over with the arsenal of your choice, and persuade anyone paying a visit to dance according to your tune.

5.1 Caveat

The presented pattern is complementary to the principal defences, not a substitute. Its worth is in preparing for the attack to come; a stratagem to make the intruder to reveal himself, his resources, methods and intentions. The pattern is useful for cases where watertight protection cannot be achived. In such cases, however :

- It does not remove the need to have the other means to counter whatever havoc the malefactor has in mind to play.

- The protection will be improved, but it still will not be unshakable.

- Presenting a weakness can direct the attack towards it; care will be needed to avoid this weakness to be successfully exploited by the attackers.

5.2 Application

5.2.1 Generic

An aspect facilitating the universal use of the pattern is that every site, however strong, does have a weakest point, as exemplified by the European medieval castles mentioned earlier; with them it was the gate, but even this weak point was very strong indeed [23]).

Also, “To catch a thief, you must think like a thief”:

First, to make the behaviour of your unwanted guest a little more predictable, you must step into the “boots” of the potential malefactors. You must take care to present something that will hopefully suit their appetite, but nothing too obvious, since a sophisticated intruder might be alerted when encountering a “too easy” target (In computer setting, a “low hanging fruit”), and decide to resort to something more intricate. Also, after a break-in, it should not be easily deductible that a trap is closing.

There are at least the following possibilities:

- Determine existing weaknesses in the defences. In case that there is no cure available (yet) these should be definitely taken in account.
- Open your defences slightly, so as not to arouse suspicion, and at a select point, so that you will know in advance the likely ingress point of an attack. This is an obviously an option that requires preparedness.
- In addition, the “gap” can be a fake. Say, by reporting misinformation about the “fort”; like location of an opening where no exists can entice an attack devised to make use of it, leading to an easy detection (with computers e.g. a wrong version of a certain software, one known to present a particularly popular vulnerability)

The presented “gap” should be irresistible to the intruder, yet easy to guard. Usually, an intruder hopes to gain the control of some central part, to allow him to investigate the surroundings and to gain foothold in as many other parts as possible. The ones which seem to be the “hubs” of any kind of traffic are natural picks.

Creating “a gap” means lowering your defences or tolerating some known vulnerabilities. And/or at least (falsely) exhibiting them. This means that there will be a certain element of risk involved, since malevolent activity will be allowed some leeway. It is up to the defender to see that he can keep his insidious guests within bounds.

The presented solution requires that the “fort” needs to be well armed and manned; in the handling of the unprecedented no automatic mechanisms has proven to be satisfactory (so far). In other words, the pattern requires substantial effort to deploy, maintain and operate.

Finally, it cannot be stressed too much that the other, principal security mechanisms must be in top condition, the “gap” being just an addition.

5.2.2 Computers

Generally, the “first line” of defence cannot be trusted to withhold a sufficiently determined attack. For example, in network security the defence is based on the models of attacks known beforehand, or detection of anomalies. So it is within possibilities that novel variants of attacks could appear, not deviating too much from the normal traffic to cause an anomaly alarm to be raised. Also, defence might fail by misconfiguration and too numerous “false positives”, the filtering of which can throw some real alerts aside as well.

The “gap” can be presented in a special “bait” machine, external to the production system, as is the case with most “honeypots”. A machine can be too obviously a “low hanging fruit”, though. A double bluff is also a possibility: A production system truly residing in an overtly tempting looking host (but still well guarded), and a somewhat less enticing target offered by some nearby machine.

Second, be aware of the prime targets of the “black hats” potentially assaulting your system.

Guard all the above keenly, but opaquely.

If a malefactor gains an access into the system through your “gap”, try to confuse him; say, let him believe that he’s in a honeypot. Or present a “forced deal” to him (e.g. “honeytokens” [11]). When deliberately offering a target you must thread carefully so that the systems intended to protect you will not be made to turn against you, or if this happens, they should not be able to cause excessive harm. This is a very real threat, since a takeover of the protective systems is the dream of a cracker.

Although a bait would be a bit too evident for a intruder, his less sophisticated colleagues will still be trapped, though, and if there is no cure for his next move, nothing additional will be lost (the system will be lost with or without implementing this pattern). Also, the “black hat” might think it’s a case of a “double bluff”.

So, while in operation, a good decoy should seem to be a realistic one, with authentic looking traffic. As such, offering and watching over “baits” is not something that a small site could consider, so a balance needs to be stricken between the needed effort and the risks (risk = probability of an incident multiplied by its impact).

Collect incriminating, non-repudiable evidence. Logging time-stamped security events to a non-rewritable medium is a good start.

5.3 Known Uses

5.3.1 Discovery of the dtspcd exploit

First documented capture of an unknown exploit with a “Honeypot” (from the “Honeynet” project [10], declared as the first in [16],). was noted in CERT advisory CA-2002-01[14]

In this document the network traces provided by the Honeynet Project were specified as the source material that revealed the active exploitation of the dtspcd vulnerability.

The vulnerability in question was a remotely exploitable buffer overflow, affecting the Common Desktop Environment’s (CDE, a graphical user interface on *nix) Subprocess Control Service network daemon (dtspcd) that accepts requests from clients to execute commands and launch applications remotely.

The details of the capture of the exploit, including the actions of the intruder, can be found in [13].

In short, the traffic of the Honeypot, a machine with SunOS 5.8, was registered with the Ethereal tool and the attack itself followed the usual procession of reconnaissance -> exploit -> reinforcement (bring in the rootkit)-> consolidation (use newly installed back-doors for further communications) as outlined in [15] .

5.3.2 Project Honey Pot

Project Honey Pot [17] is an on-going and operational (at the time of writing, March 2009) effort to snare e-mail address harvesters, spammers, dictionary attackers and comment spammers [18]. “Harvesters” are automated collectors of addresses from web-pages, “dictionary attackers” are in search of e-mail addresses by mailing to potential user names and waiting for possible acknowledgements, and “comment spammers” are targeting blogs and forums. The idea of the Honey Pot is to present decoy e-mail addresses and html forms in www-pages. When these (unique) e-mail addresses are “harvested”, the “harvester’s” IP-address will be stored with the harvested address. Immediately after this the decoy e-mail address will be changed. The arrangement facilitates later correlation of the harvesters IP-address with the email sent by a spammer to this address (the spammer and harvester will have most probably a “subcontractor” relationship, and therefore a different IP address).

An noteworthy aspect of the activity is that the system is global, orchestrated by the Project Honey Pot that publishes statistics and results, cooperates with law enforcement officials and accepts donations. The donations might be e.g. MX records that can be used in decoys and www-pages in servers along with plain funding.

5.3.3 Tom Liston's "LaBrea" Tar Pit vs. "CodeRed" worm

The "CodeRed" worm [20] was observed July 13 2001 and Tom Liston's site was one of the victims. He came to think about ways to slow it down, and settled on offering bogus machines to the worms entering reconnaissance phase, that is, starting to find new victim machines by scanning. The bogus machines were announced at the communication protocol level (starting with ARP and ending with TCP) to any request to connect to an address with no machine attached. Since the protocols at the worm's side were persistent about opening a connection to the new found "virtual machine", substantial delays could be effected by careful choice and timing of the protocol replies. When the worm finally moved on (to the next address), the same procedure could be repeated. [19]. The fruition of this idea was "LaBrea" [1].

5.4 Exemplary cases

The pattern has a wide area of application:

- You might run a networked site, with connections available for external traffic. Most often you have to provide services (like web) with an indeterminate set of external users/machines so a VPN (Virtual Private Network) solution is not applicable. So, you need to fend against the threat from the "outside". As for the remedy, the solution might include "Honeypots", "Tarpits" and Logging arrangements, see 5.5

- Another case could be a user interface provided for the public, say a desktop system in libraries, embedded systems for payments like at petrol stations or ATMs (Automatic Teller Machines). A potential intruder might try to sneak his/hers way around the official user interface. In this case the intruder might be "rewarded" for his persistence, by having sufficiently hidden items like "system control" or "authorised users only" to be available to him or her, triggering appropriate alarms and video cameras etc. as a matter of course. The items should have varying (preferably random) discovery "paths" and announcements to prevent their learning.

- Also within singular applications, like databases, users apt for unauthorised access need to be trapped. Here suitable "Honeytokens" [11], e.g. extra tables with tempting looking content (fake credit card numbers etc.) but with access announced as illicit to ordinary users might be the answer.

- The pattern applies beyond computerised systems, say to physical buildings where an attempt needs to be made to oversee the comings and goings of people. In buildings, a special, unlocked door, with alarms, and with a sign like "Cashiers office, for personnel only" could do the trick,

- And, if you ever find yourself in a situation calling for the planning of the fortifications for a village, in circumstances resembling medieval Japan, you can also take the name of the pattern quite literally (see 5.5)

5.5 Suggested Further Reading

5.5.1 On Honey Things

Honeypots: Usually thought to be separate machines, but the idea fits to separate entities within machines as well (processes, virtual machines). These are not so effective anymore, since these can be detected [3][2], even easily[5], since they need to abide with the law and it is trivial to test this. Simulated "unlawful" environments might be an improvement as proposed in [5], but are not likely to be an answer since the attackers can test against several known (to them) domains, and not all of them can be fakes. Undoubtedly, the (counter) detection methods will become more sophisticated and/or inventive with time. Aside the simulated unlawfulness, the "honeypots" need to be made to resemble normal systems as closely as possible, including fake traffic with normal looking traffic patterns.

Honeytokens: Associated with honeypots are “honeytokens”, i.e. a piece of information used as a bait, essentially something that the intruder can “take away” and that can act after its capture as a means of tracking of the actions taken by the intruder. As noted in [11], the honeytoken can be made to incorporate also the aspect of “agent provocateur”, i.e. to contain something capable of enticing its capturer to commit subsequent, incriminating actions (say, an e-mail address, with a promise of extra information, given in the honeytoken only, for the “special guest”, could tempt the “black hat” actually to send a message and thus reveal himself).

Honeynets: Comprise of several connected honeypots, see [10]

Honeymonkey: Of Microsoft origin [7], the idea is to actively invite marauders into your system. Honeymonkey (there are also open source efforts, “honeyclient” and “HoneyC[8][9]) contacts websites, downloads their content and checks afterwards whether any “monkey business” (or worse) was perpetrated.

5.5.2 On Parallel Approaches

Tarpits: “Tarpits” [1] are based on the idea of slowing down sessions to deter intruders. In case of “LaBrea”[1] the main idea is to offer empty promises of hosts to contact, kind of “virtual” honeypots, so that the assailants attempts to reach the non-existent will waste his/hers time.

***nix syslog:** The logs can be periodically and silently (unseen by the rest of the system) stored in a secured place for (sufficiently frequent) periodic comparison. Logs being a valued target, attempts to cover traces of illicit activity are thus easily detected. This idea is described in [3], and its nature is a “last resort” affair, since it implies that all preceding safeguards have failed.

6 Resulting Context

A warning has been received and the “cat and mouse” play between the intruder and the admin may begin.

Since it is increasingly difficult to get any reliable information about the intruders’ ultimate traffic source (series of “bots” used for mediating), the malefactor should be kept unaware of his/hers exposure, so that useful information about the “modus operandi” can be accumulated. This, and not the simple act of filtering packets based on source addresses, can be used to prevent any subsequent similar attempts in the future.

There are three main cases:

1. The “mouse” continues to believe in the hoax, and after performing whatever it had in its mind, leaves. In a lucky case it might further contribute to its downfall, say, in a computer setting, a “honeytoken” arouses its interest so it scurries away with it between it’s teeth.

2. After a while, the “mouse” begins to detect that something is amiss, retreats, maybe after an attempt to conceal any telltale signs, and lists the “gap” as intentional. Perhaps it distributes this information to its kind.

3. The “mouse” was specifically in search of defence related arrangements, or was otherwise prepared to handle the situation, and is determined to make use of them.

As for the “cat”, in the first case the follow-up actions are to analyse the evidence left behind to learn the tricks of the intruder, implement any necessary improvements, and if appropriate, wait for “hooks” in a possibly swallowed honeytoken to catch, or the honeytoken itself to show up.

The second case will lead to a need for further actions, since the plot has been revealed. The attacker might ponder returning with new vigour at some future date, or some of the kindred souls that have received or bought the information might pay a visit, avoiding or targeting the "gap" as they wish.

With the third case great care is needed, to avoid the bait to turn against its masters. If the going gets touch the gatecrasher can be forcibly expelled to avoid him or her gaining any foothold. In any of the cases, after an noticed attack attempt, thorough checking of the relevant systems is commendable.

7 Origins of the idea

The basic insight has ancient roots; as it happens, the principle was used in medieval Japanese warfare. According to Karl F. Friday, a professor in Japanese history, the early fortifications (12th - early 13th century) ubiquitously employed wooden gates, *kido* or *kidoguchi*, designed to act as focal points of a battle, since they were the only conceivable entrances to the attackers (as well as a means of counterattack to the defenders) of the time [12].

The idea is also depicted in the Kurosawa's famous film "The Seven Samurai" (1954). The name chosen for this pattern is taken from a line of one of the main characters of that movie, in a scene where the construction of the defences for a village threatened by marauders is pondered: "A good fort needs a gap. The enemy must be lured in. So we can attack them. If we only defend, we lose the war."[6].

Also, Japanese castle town's layout often featured winding streets and blind alleys designed to delay and deceive the attackers [21] .

8 Acknowledgements

It was Mr Juha Pärssinen who suggested to me to write some patterns, an exercise I have not tried before. While I have been aware of patterns and have read a bunch of them, it was his enthusiasm and advice that prompted me to produce the current text. I thereby present my thanks to Juha for guiding me in the world of patterns.

I also owe thanks to my "shepherd", Kristian Elof Sørensen for many valuable suggestions concerning the form and the content of the pattern. And, I am gratefull for Tim Wellhausen, Ville Reijonen, Veli-Pekka Eloranta, Marko Leppänen, Anjali Das, Farah N. Lakhani, Martin Wagner and Stefan Sobernig for offering many heplfull comments during the EuroPloP-09 conference.

9 Copyright

References

[1] A tarpit: http://labrea.sourceforge.net/labrea-info.html (retrieved 13. Nov 2009)

[2] http://www.send-safe.com/honeypot-hunter.html (retrieved 13. Nov 2009)

[3] Anti-honeypot technology, Security & Privacy Magazine IEEE, http://ieeexplore.ieee.org/iel5/8013/28290/01264861.pdf?arnumber=1264861 (retrieved 13. Nov 2009)

[4] On logging http://www.sans.org/reading_room/whitepapers/logging/1168.php (retrieved 13. Nov 2009)

[5] Detecting honeypots, the easy way http://www.springerlink.com/content/dq3t9btumw6gvrk5/ (retrieved 13. Nov 2009)

[6] http://en.wikiquote.org/wiki/The_Seven_Samurai (retrieved 13. Nov 2009)

[7] ftp://ftp.research.microsoft.com/pub/tr/TR-2005-72.pdf (retrieved 13. Nov 2009)

[8] http://www.honeyclient.org/trac (retrieved 13. Nov 2009)

[9] https://www.client-honeynet.org/ (retrieved 3. Feb 2009)

[10] https://www.honeynet.org/ (retrieved 13. Nov 2009)

[11] http://www.securityfocus.com/infocus/1713 (retrieved 13. Nov 2009)

[12] Samurai, Warfare and the State in Early Medieval Japan, by Karl F. Friday, Published by Routledge, 2004, ISBN 0415329620, 9780415329620 (Chapter "Fortifications and Strongholds", page 125).

[13] http://old.honeynet.org/scans/scan28/sol/29/sotm28.pdf (retrieved 13. Nov 2009)

[14] http://www.cert.org/advisories/CA-2002-01.html (retrieved 13 Nov 2009)

[15] Richard Bejtlich, The Tao of Network Security Monitoring: Beyond Intrusion Detection, ISBN-10: 0-321-24677-2, ISBN-13: 978-0-321-24677-6

[16] Honeypots: Tracking Hackers, by Lance Spitzner, Publisher: Addison-Wesley Professional, Pub Date: September 10, 2002, ISBN-10: 0-321-10895-7, ISBN-13: 978-0-321-10895-1

[17] http://projecthoneypot.org/index.php

[18] http://projecthoneypot.org/faq.php (retrieved 13. Nov 2009)

[19] http://labrea.sourceforge.net/Intro-History.html (retrieved 13. Nov 2009)

[20] http://en.wikipedia.org/wiki/Code_Red_(computer_worm) (retrieved 13. Nov 2009)

[21] http://en.wikipedia.org/wiki/Japanese_castle#Layout (retrieved 13. Nov 2009)

[22] Jurgen Brauer and Hubert van Tuyll: Castles, Battles, and Bombs: How Economics Explains Military Historyl'2008, ISBN: 978-0-226-07163-3 (ISBN-10: 0-226-07163-4), pages 45–66

[23] http://en.wikipedia.org/wiki/Castle#Gatehouse (retrieved 13. Nov 2009)

[24] http://upload.wikimedia.org/wikipedia/commons/a/a9/Old_painting_of_Himeji_castle.jpg (retrieved 13. Nov 2009)