E-government security: A honeynet approach

[1]Bahman Nikkhahan, [2]Akbar Jangi Aghdam, and [3]Sahar Sohrabi
[1]*K. N. Toosi University of Technology of Iran, bahman616@gmail.com*
[2]*Iran University of Science and Technology, a.aghdam@gmail.com*
[3]*K. N. Toosi University of Technology of Iran, shr.sohrabi@gmail.com*

Abstract

Security is one of the most important issues in E-government. All of the security approaches that are common in E-commerce are applicable to E-government. But E-government is a little different from E-commerce. Usually government networks can communicate to each other better than business networks, because, most of them are connected for transferring information, but businesses are competitors and they don't disclose their sensitive information. Utilizing "honeypots" is a good solution for tracing hackers and revealing their tools. In this paper, "connectedness" of E-government networks and honeypots are employed to propose an approach for securing an E-government network. This framework provides suitable resources for hackers; and simultaneously, it prevents them from misusing those resources for future attacks.

1. Introduction

Information and Communication Technologies (ICT) is transforming the governmental processes in serving citizens (G2C), businesses (G2B) and governments (G2G).

While E-government is subject to the same threats as e-business, E-government operates within different constraints. Most businesses deal only with a subset of the population, and they can choose the how and the when they do it. But the government must deal with everyone [10]. Therefore, in order to the huge number of users and transactions, and sensitivity of this field, like citizen's private information or government's secret information, and other issues, securing governmental networks is more important than businesses [2]. One of the main issues of trust in E-government implementation is security [11]. Citizens prefer to use traditional ways rather than using an unsecured web site. On 14 June 2002 the UK’s Inland Revenue withdrew its online tax filing service amid complaints that users could see other people’s tax returns. This public humiliation, however temporary, reveals part of the price paid when E-government initiatives are not secure [10].

Henriksson et al. (2006) divided the factors that influence the quality of government websites to 6 major categories: (1) Security and Privacy; (2) Usability; (3) Content; (4) Services; (5) Citizen Participation; and (6) Features [4].

Wimmer and Bredow (2001) proposed a holistic concept that integrates security aspects from the strategic level down to the data and information level in order to address different security aspects of E-government in a comprehensive way. Their holistic approach consists of 4 layers: strategic, process level, interaction and information [12].

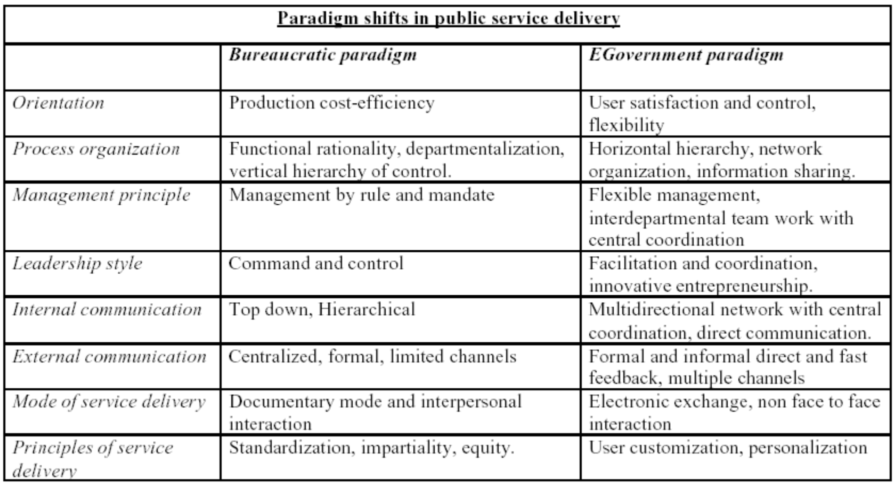
Hof and Reichstädter surveyed security, peculiarities and implementations of security requirements within governmental structures, based on three interaction points (citizen to government C2G, government to government G2G and government to citizen G2C) [5].

This paper introduces a fault tolerance honeynet to strengthen the security of governmental network. At first, we will describe E-government, honeypots and honeynet in detail, and then we will show the proposed framework.

2. E-government

The initiatives of government agencies and departments to use ICT tools and applications, Internet and mobile devices to support good governance, strengthen existing relationships and build new partnerships within civil society, are known as eGovernment initiatives (see table 1). As with e-commerce, eGovernment represents the introduction of a great wave of technological innovation as well as government reinvention. It represents a tremendous impetus to move forward in the 21st century with higher quality, cost effective government services and a better relationship between citizens and government [6].

Table 1. Reinventing Local Governments and the E-government Initiative [6]



Paradigm shifts in public service delivery

	Bureaucratic paradigm	*EGovernment paradigm*
Orientation	Production cost-efficiency	User satisfaction and control, flexibility
Process organization	Functional rationality, departmentalization, vertical hierarchy of control.	Horizontal hierarchy, network organization, information sharing.
Management principle	Management by rule and mandate	Flexible management, interdepartmental team work with central coordination
Leadership style	Command and control	Facilitation and coordination, innovative entrepreneurship.
Internal communication	Top down, Hierarchical	Multidirectional network with central coordination, direct communication.
External communication	Centralized, formal, limited channels	Formal and informal direct and fast feedback, multiple channels
Mode of service delivery	Documentary mode and interpersonal interaction	Electronic exchange, non face to face interaction
Principles of service delivery	Standardization, impartiality, equity.	User customization, personalization

E-government means different things for different people. Some simply define it as digital governmental information or a way of engaging in digital transactions with customers. For others E-government simply consists of the creation of a web site where information about political and governmental issues is presented. These narrow ways of defining and conceptualizing E-government restrict the range of opportunities it offers [6].

Different authors have different definition of E-government:

Richard Heeks propose that the term "E-governance" should be seen to encompass all ICTs, but the key innovation is that of computer networks, from intranet to the Internet, which have created a wealth of new digital connections:

- Connections within government, permitting "joined-up thinking."
- Connections between government and NGO/citizens, strengthening accountability.
- Connections within and between NGOs, supporting learning and concerted action.
- Connections within and between communities, building social and economic development.

As a result, Heeks suggest, the focus of e-governance shifts from just parts of e-administration, in the case e-government, to also encompass e-citizens, e-services and e-society [3].

- Whitson and Davis (2001): "Implementing cost-effective models for citizens, industry, federal employees, and other stakeholders to conduct business transactions online".
- Tapscott (1996): “An inter-networked government”.
- Luling (2001): “online government services, that is, any interaction one might have with any government body or agency, using the Internet or the World Wide Web” [8].

Inter-networked government is the best definition for the purpose of this paper

3. Honeypots

Honeypots are a security resource whose value lies in being probed, attacked or compromised. This means that whatever we designate as a honeypot, our expectations and goals are to have the system probed, attacked, and potentially exploited. It does not matter what the resource is (a router, scripts running emulated services, a jail, an actual production system). What does matter is that the resource's value lies in its being attacked. If the system is never probed or attacked, then it has little or no value. This is the exact opposite of most production systems, which you do *not* want to be probed or attacked [9].

The primary purpose of a honeypot is to proactively gather information about security threats by providing a real system with real applications and services for the attacker to interact with, but with no production value: we can safely watch and learn from an intruder without fear of compromising our systems [1]. The value of a honeypot is weighed by the information that can be obtained from it [7].

Traditionally, the attacker has always had the initiative. They control whom they attack, when, and how. All we can do in the security community is defend: build security measures, prevent the bad guy from getting in, and then detect whenever those preventive measures fail. As any good military strategist will tell you, the secret to a good defense is a good offense. But how do the good guys take the initiative in cyberspace? Security administrators can't go randomly attacking every system that probes them. We would end up taking down the Internet, not to mention the liability issues involved. Organizations have always been limited on how they can take the battle to the attacker. Honeypots give us the advantage by giving us control: we allow the bad guys to attack them [9].

Honeypots can run any operating system and any number of services. The configured services determine the vectors available to an adversary for compromising or probing the system [7].

Honeypots are categorized by the level of interaction into high-interaction and low-interaction.

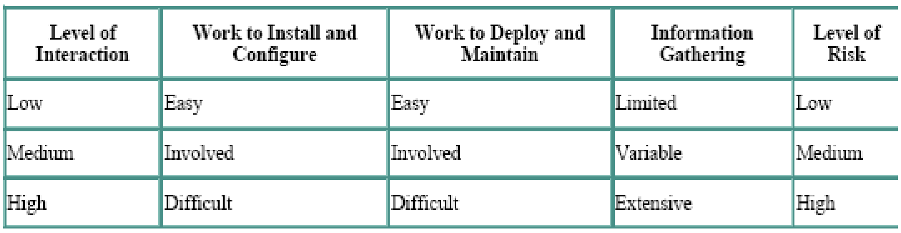
Level of interaction gives us a scale with which to measure and compare honeypots. The more a honeypot can do and the more an attacker can do to a honeypot, the greater the information that can be derived from it. However, by the same token, the more an attacker can do to the honeypot, the more potential damage an attacker can do [9].

A high-interaction honeypot provides a real system the attacker can interact with. It can be compromised completely, allowing an adversary to gain full access to the system and use it to launch further network attacks.

In contrast, a low-interaction honeypots simulates only some parts — for example, the network stack. These honeypots simulate only services that cannot be exploited to get complete access to the honeypot. A low-interaction honeypot often implements just enough of the Internet protocols, usually TCP and IP, to allow interaction with the adversary and make her believe she is connecting to a real system [7].

Whether you use a low-interaction or high-interaction honeypot depends on what you want to achieve. Table 2 summarizes the tradeoffs between different levels of interaction in four categories [9].

Table 2. Tradeoffs of honeypot levels of interaction [9]



Level of Interaction	Work to Install and Configure	Work to Deploy and Maintain	Information Gathering	Level of Risk
Low	Easy	Easy	Limited	Low
Medium	Involved	Involved	Variable	Medium
High	Difficult	Difficult	Extensive	High

The first category is installation and configuration, which defines the time and effort in installing and configuring your honeypot. In general, the greater the level of interaction a honeypot supports, the more work required to install and configure it. This is simply common sense. The more functionality you provide an attacker, the more options and services must be installed and configured.

The second category is deployment and maintainance. This category defines the time and effort involved in deploying and maintaining your honeypot after you have built and configured the system. Once again, the more functionality your honeypot provides, the more work required to deploy and maintain it.

The third category is information gathering—how much data can the honeypot gain on attackers and their activities? High-interaction honeypots can gather vast amounts of information, whereas low-interaction honeypots are highly limited.

Finally, level of interaction impacts the amount of risk introduced. We are concerned about the risk of a honeypot being used to attack, harm, or infiltrate other systems or organizations. The greater the level of interaction, the more functionality provided to the attacker, and the greater the complexity. Combined, these elements can introduce a great deal of risk. On the other hand, low-interaction honeypots are very simple and offer little interaction to attackers, creating a far lower risk solution[9].

We also differentiate between physical and virtual honeypots.

3.1. Physical honeypot

Physical honeypot means that the honeypot is running on a physical machine. Physical often implies high-interaction, thus allowing the system to be compromised completely. They are typically expensive to install and maintain. For large address spaces, it is impractical or impossible to deploy a physical honeypot for each IP address. In that case, we need to deploy virtual honeypots [7].

3.2. Virtual honeypot

Compared to physical honeypots, this approach is more lightweight. Instead of deploying a physical computer system that acts as a honeypot, we can also deploy one physical computer that hosts several virtual machines that act as honeypots. This leads to easier maintenance and lower physical requirements. Usually VMware or User-Mode Linux (UML) are used to set up such virtual honeypots. These two tools allow us to run multiple operating systems and their applications concurrently on a single physical machine, making it much easier to collect data [7].

3.3. Advantages and disadvantages of various kinds of honeypots

With the help of a high-interaction honeypot, we can collect in-depth information about the procedures of an attacker [7]. We can watch how she attacks and what kinds of tools and approaches she uses.

High-interaction honeypots — both virtual and physical — also bear some risks. In contrast to a low-interaction honeypot, the attacker can get full access to a conventional computer system and begin malicious actions. For example, she could try to attack other hosts on the Internet starting from your honeypot, or she could send spam from one of the compromised machines [7].

Low-interaction honeypots can be used to detect known exploits and measure how often your network gets attacked. The advantages of low-interaction honeypots are manifold. They are easy to set up and maintain. They do not require significant computing resources, and they cannot be compromised by adversaries. The risk of running low-interaction honeypots is much smaller than running honeypots that adversaries can break into and control. On the other hand, that is also one of the main disadvantages of the low-interaction honeypots. They only present the illusion of a machine, which may be pretty sophisticated, but it still does not provide an attacker with a real root shell [7].

One disadvantage of virtual honeypot is the attacker can differentiate between a virtual machine and a real one. It might happen that an advanced attacker compromises a virtual honeypot, detects the suspicious environment, and then leaves the honeypot again. Moreover, she could change his tactics in other ways to try to fool the investigator. So virtual honeypots could lead to less information about attackers [7].

3.4. Honeynet and Honeywall

Honeynet is a group of linked honeypots behind a special firewall called a honeywall [1]. Usually, a honeynet consists of several honeypots of different type (different platforms and/or operating systems). This allows us to simultaneously collect data about different types of attacks. Usually we can learn in-depth information about attacks and therefore get qualitative results of attacker behavior [7].

Also, the Honeywall is normally set up as a transparent bridge that limits the amount of malicious traffic that can leave the honeynet, keeping an attacker from attacking other machines on the Internet [1].

4. The proposed model of a Fault tolerance honeynet for securing e-government

Securing E-government networks is similar to other networks. Many approaches like cryptography, PKI, firewalls, digital signatures are employed in these networks. However, as mentioned above, E-government is an inter-networked government. In most of the cases, government agencies in a country are connected to each other for communicating the information about citizens. This is one of the main differences between government networks and business networks. Because, the businesses are competitor and do not disclose their network to each other, but in most of the governments, co-operation is more critical than competition. So we can use this connectedness to set up a honeynet.

The main goal of honeypots is to trace the hackers and obtain information about their approaches and tools. One of the most important challenges of honeypots is the degree of their interaction. If we use low-interaction honeypots, a hacker cannot utilize all of the resources of the system, so she probably won't be able to use all of her approaches and tools. Therefore we will lose a suitable opportunity for obtaining information. In the other side if she can completely utilize all of the resources, maybe she can use the information that she obtained from the honeypot for attacking other hosts or send spam from one of the compromised machine [7]. As a result, this trade off must be managed effectively.

Security in the governmental networks is more critical than business networks. So if we want to use honeypots in these networks, we must consider the trade off related to interaction, precisely. We need at least a high-interaction honeypot for each agency's network. So we have a network of honeypots through the government, called honeynet. With this network we can increase the possibility of attacks; because our honeypots are dispersed all over the government and all of them are high-interaction. Furthermore we have a Honeycentre server. This server is the manager of the honeynet. It aggregates all of the honeypots logs and then summarizes the results. Honeycentre then informs the web servers about the results, so the administrators of those web servers make an appropriate defensive decision to cover the security holes.

Now we have a network that traces the attacks, all over the government, with a high degree of interaction to hackers. On the other hand, we can cover our security holes as soon as possible.

Honeypots are subject to damage. So, various attacks may disable the honeypots. So we must have a fault tolerance network to predict these problems and react as soon as possible. For this purpose, Honeycentre can help. Honeycentre is gathering information and logs from honeypots all of the times. When a honeypot is down, Honeycentre cannot receive the logs from that honeypot, so it informs the web server of that network and simultaneously assigns a virtual honeypot instead of the damaged honeypot. Honeycentre allots IP address of the damaged honeypot to the virtual honeypot. We may have an additional server for assigning these virtual honeypots (figure1).

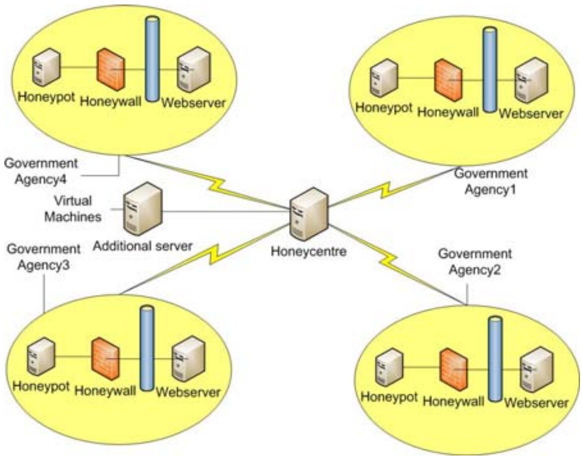


Figure 1. Fault tolerance honeynet for securing e-government

In figure 1, other network components like user stations, gateways, routers, and connections between agencies are not shown for avoiding complexities.

As mentioned in section 2.2, virtual honeypots are more lightweight than a high-interaction or low interaction honeypot. At a given time, we may have some damaged honeypots in the network, and we cannot fix all of them very soon. In the other hand we cannot assign some high or low interaction honeypot instead of all damaged honeypots to the network, because maybe they are too many and we don't have required resources. So if we use virtual honeypots temporarily, we can solve this problem with just one additional server.

These virtual honeypots must act like the original honeypot. So each government agency replicates its minimum data into Honeycentre or additional server in every specified period of time. These data are minimum, because they should only help the system work, until the problem is solved and the real honeypot returns back to its logical position.

Such a situation enables the Honeycentre to create a real fault tolerant system which would be strong enough to deal with attacks without any interruption.

Figure 2 shows this process in the Honeycentre. Honeycentre wants to gather data from all of the honeypots; for this purpose, Honeycentre checks responses from all of the honeypots. If a honeypot did not respond in a specific period of time, Honeycentre finds out a problem with that honeypot. So, it sends an error report to the administrator of the network, then allocates a virtual honeypot for that network and finally updates honeylist. Honeylist consists of addresses of all honeypots and their related web servers and administrators of their networks. If the honeypot responds, Honeycentre downloads data about hackers from the honeypot. Then it downloads required governmental data into additional server for running virtual honeypot, if needed.

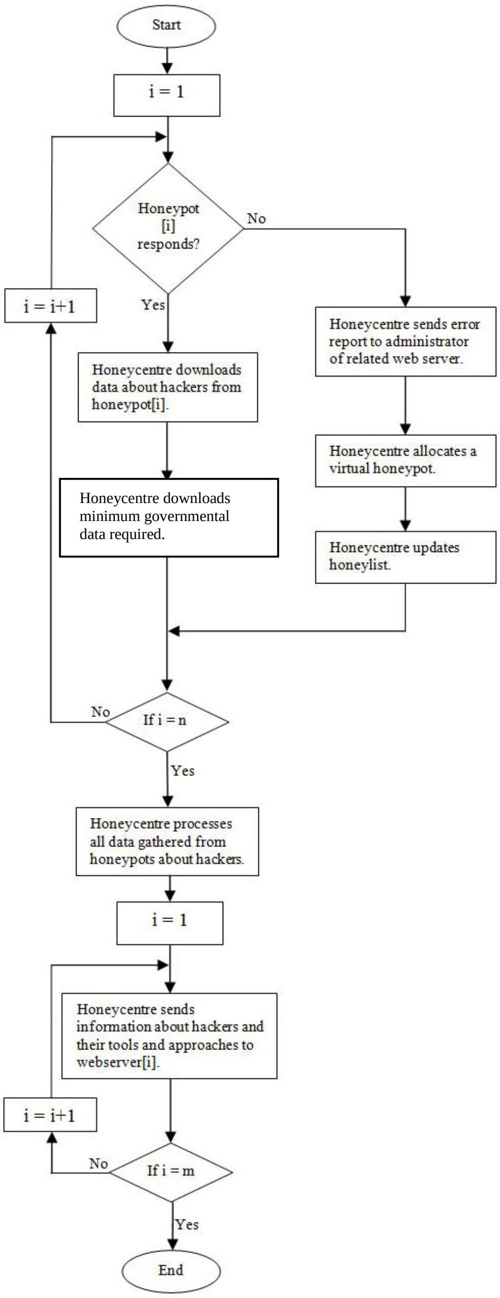

Start
i = 1
Honeypot [i] responds?
No
Yes
i = i+1
Honeycentre downloads data about hackers from honeypot[i].
Honeycentre downloads minimum governmental data required.
Honeycentre sends error report to administrator of related web server.
Honeycentre allocates a virtual honeypot.
Honeycentre updates honeylist.
No
If i = n
Yes
Honeycentre processes all data gathered from honeypots about hackers.
i = 1
Honeycentre sends information about hackers and their tools and approaches to webserver[i].
i = i+1
No
If i = m
Yes
End

Figure 2. The flowchart of the framework in Honeycentre

When Honeycentre collected the data from all honeypots, processes them and converts these data to useful information. Then, it sends information to all of the web servers. This information consists of approaches and tools that hackers employ and their anti hack solutions.

With this framework, we can consider the interaction trade off, that mentioned above, effectively. From one point of view, we utilize high interaction honeypots and from other point of view, we create boundaries to prevent hackers from doing more than their permission.

5. Conclusion

Designing and implementing more effective approaches for securing E-government is an important issue, because, the governmental information is usually so sensitive. Furthermore, security has an important role in trust formation of citizens and their adoption of e-government. In this paper, one of the main differences of government and business networks is exploited: connectedness. This useful property used to form a network of honeypots. Furthermore, this honeynet is fault tolerance; so if some honeypots are damaged, the Honeycentre allots some virtual honeypots with minimum resources needed, and evicts the damaged honeypot from the network. So the proposed framework causes interaction with the hackers completely and simultaneously prevents them from damaging the network.

References

[1] Barfar, A. and Mohammadi, S., 2007, Honeypots: Intrusion deception, ISSA Journal, pp. 28-31.

[2] Conklin, A. and White, G. B., 2006, E-government and Cyber Security: The Role of Cyber Security exercises, Proceedings of the 39th Annual Hawaii International Conference on System Sciences.

[3] Fallahi, M., 2007, The obstacles and guidelines of establishing E-government in Iran, MSc. Thesis, Luleå University of Technology, Sweden, available online at: http://epubl.ltu.se/1653-0187/2007/052/LTU-PB-EX-07052-SE.pdf

[4] Henriksson, A. Yi, Y. Frost, B. and Middleton, M. 2006, Evaluation instrument for e-government websites, Proceedings Internet Research 7.0: Internet Convergences, Brisbane, Queensland, Australia.

[5] Hof, S. and Reichstädter, P., 2004, Securing e-Government, EGOV 2004, Springer-Verlag, 2004, pp. 336-341.

[6] Ndou, V., M., 2004, E-government for developing countries: opportunities and challenges, The Electronic Journal on Information Systems in Developing Countries, 18(1), pp. 1-24.

[7] Provos, N. and Holz, T., 2008, Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Addison Wesley Professional.

[8] Sharifi, H. and Zarei, B., 2004, An adaptive approach for implementing E-government in I.R. Iran, Journal of Government Information, 30(5-6), pp. 600-619.

[9] Spitzner, L., 2002, Tracking Hackers, Addison Wesley Professional.

[10] Stibbe, M., 2005, E-government security, Infosecurity Today, 2(3), pp. 8-10.

[11] The E-government handbook for developing countries Infodev, 2002.

[12] Wimmer, M. and von Bredow B., 2001, E-Government: Aspects of Security on Different Layers, Proceedings of the 12th International Workshop on Database and Expert Systems Applications (DEXA’01).

Authors

Bahman Nikkhahan is a master student of IT engineering in the Khajeh Nasir Toosi University of Technology (KNTU), Iran. He received his B.S. in hardware engineering from Iran University of Science and Technology(IUST). He also is an IT project manager in the New Data Age company. His main research interests is software engineering, E-government, E-commerce and network security. He may be reached at bahman616@gmail.com.

Akbar Jangi Aghdam received his B.S. in computer hardware engineering from Iran University of Science and Technology(IUST). He is an IT project developer and sales manager in the New Data Age company. His main research interests is managment, E-government, network-marketing, E-commerce and computer architecture and networks. He may be reached at a.aghdam@gmail.com.

Sahar Sohrabi received his B.S. in computer science from Tabriz university, Iran, in 2006. She is studying M.Sc. in Khajeh Nasir Toosi University of Technology(KNTU) now. Her research interests include mathematical modeling, and heuristic and metaheuristic optimization methods. She may be reached at shr.sohrabi@gmail.com.