GeoBureau
Geographic Information Specialists

Ordnance Survey
Developer Partner

ESRI
AUTHORIZED DEVELOPER

Security Analysis of Micro-payment Systems

Stilianos Vidalis

Information Security Consultant
Geo-Bureau Ltd, 47 Cowbridge Road,
Pontyclun, Rhondda Cynon Taf,
UK, CF72 9EB
e-mail: stilianos.vidalis@geobureau.co.uk
Tel: +44 (0) 845 603 10 10
Fax: +44 (0) 1443 48 23 29

Table of Contents

List of Figures

List of Tables

Abstract

In this report we will examine the various Micro-Payment Systems (MPS) that are currently defining the state of the art in the banking and other financial sectors. MPSs (Donal O'Mahony '97) have been developed in the past decade, but for various reasons (Shirky '00) have not yet penetrated the market. The wide development of the mobile Internet though, is creating an opportunity to change that and already a lot of companies are starting to use such systems for providing services. The critical mass of users will probably be reached very soon in some industrial countries where Internet has a deep penetration. This has an effect in the fragile balance between the defenders and the offenders of computing systems. There is a need to analyse this new “player” and understand the new issues that arise by the introduction of MPSs to our everyday life. After an initial analysis on the MPS technology, we will go in depth and examine the architecture and the protocols that are being used for the transactions of the MPSs. We will conclude the report by presenting a detailed analysis of the major vulnerabilities of the examined MPSs.

Micro-Payment Systems Overview

Micro-payments refer to low value electronic transactions. They provide an alternative revenue source for content providers beyond advertising and subscriptions (W3C '99). They may also provide revenue streams for service providers. Based on Thomas (Thomas '02), the m-commerce, part of which are the micro-payment systems, will be worth £2.8bn by 2005. According to Pierce (M. Pierce '97), Micro-payments involve:

- A buyer / client
- A vendor / data editor
- One or more brokers / intermediates / billing servers.

According to the World Wide Web Consortium (W3C) (W3C '99) the requirements of micro-payment systems are:

- Embedding must be simple (click & pay interface)
- Embedding must minimize TCP/IP packet round trips
- Embedding must use standard browser features

The basic architecture of a micro-payment system consist of:

- On the client side:
  - A browser,
  - A module which is communicating with the micro-payment server
  - One or more electronic wallets
- On the vendor side:
  - An HTTP server

The following are the micro-payment implementations that were examined/analyzed: PayDirect, FirstVirtual, PayItMobile, CyberCoin, QPass, Millicent, Banxafe, Bibit, Earthport, Digicash, Internet Dollar, FirstGate, Pay2See, MicroMint, Genion, CyberCent, Mobipay, Paybox. The ePSO (electronic Payment Systems Obnservatory), managed by the European Commission/Joint Research Centre/IPTS, has been used for the identification of the above MPS. These systems represent the state of the art from an international point of view (20 different countries). Details for each system can be seen in appendix A. The biggest risk of such systems is that of user acceptance. The “modern e-consumer” (see Mayes (Mayes '03)) has a lot more requirements than a traditional consumer of the ‘90s. The system must be simple and easy to use and able to earn the user’s trust. Although some of the above systems are still operational, they are far from being successful, mostly due to the failure of understanding user preferences. In agreement with Shirky (Shirky '00) micro-payment systems, typically, treat cheap resources as precious commodities, while treating the user’s time as if it were so abundant as to be free.

Micro-Payment Systems Analysis

Five digital money systems were chosen for further study. These are: CyberCash’s CyberCoin, DigiCash’s eCash, Carnegie Mellon’s NetBill, Digital’s Millicent, and METEORE 2000. These systems have provided leadership, creativity and understanding of the market’s micro-payment needs. As it will be seen they provide different solutions to the micro-payment transaction problems, hence they were chosen for this analysis. The subsequent sections characterize these schemes using the following criteria:

1. General Concept Description
2. Ease of Use
3. Anonymity & Privacy
4. Security

General Concept

- CyberCash’s CyberCoin: CyberCoin services are distributed through banks, which offers online merchants the CyberCoin service and offers consumers co-branded "Wallets." The merchants pay the bank a per-transaction fee to use the system. This fee is a tier-based pricing model based on the transaction size.

- DigiCash’s eCash: This system is based on what is called a single use token system. The user generates blinded electronic bank notes and sends them to his bank to be signed with his bank's public key (PK). The bank signs the notes, deducts the amount from the user's account, and sends the signed notes back to the user. The user removes the blinding factor and uses them to purchase at the shop. The shop verifies the authenticity of the bank notes using the bank's corresponding public key and sends them to the bank where they are checked against a list of notes already spent. The amount is deposited into the shop's account, the deposit confirmed, and the shop in turn sends out the goods. All communication over the network is protected by encryption.

  The system involves software for both the consumer and the merchant to conduct the transactions. The customer runs a "wallet" program. The user can spend the digital money at any shop accepting eCash, without the trouble of having to open an account there first, or having to transmit credit card numbers.

Because the received eCash is the value involved with the transaction, shops can instantly provide the goods or services requested.

- Carnegie Mellon’s NetBill: Once a Client approves a purchase of a good that can be transferred through the Internet, a digitally signed request is sent to the Vendor. The Vendor computes a checksum of the goods, and sends the encrypted goods with a time stamp to the Client. The Client’s software computes a checksum of the goods and then sends this, along with the accepted price, the product identifier, and the timestamp, back to the Vendor.

  After the goods have been received and verified, the Vendor’s software appends the decryption key and endorses it with the Client’s digital signature. The merchant then sends this to the NetBill server, which acts as the Broker.

  The Broker verifies that the product identifiers, prices and checksums are all in agreement. If the customer has the necessary funds or credit in his account, the Broker debits the customer's account and credits the merchant's account, logs the transaction, and saves a copy of the decryption key. The Broker then returns to the Vendor a digitally signed message containing an approval, or an error code indicating why the transaction failed. The Vendor forwards the Broker’s reply and (if appropriate) the decryption key to the Client.

- DEC’s Millicent: Millicent is a decentralized micro-payment scheme, which is designed to allow payments as low as 1/10 of a cent. It uses a form of electronic currency, which is called “scrip”. It is designed to make the cost of committing a fraud, more than the value of the actual transaction. It uses symmetric encryption for all data transactions. The principal actors of the scheme are the Broker, the Customer and the Vendor**.** Figure 1 (Source: Pierce (M. Pierce '97)) demonstrates the scheme.
-

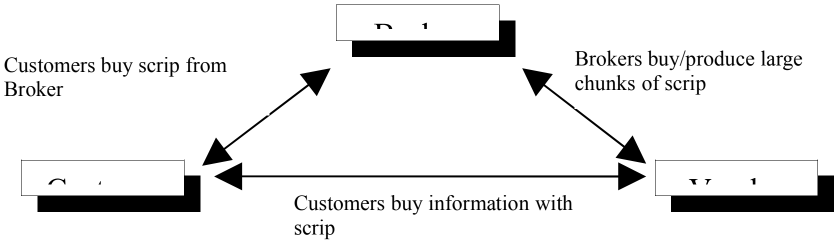
Customers buy scrip from Broker
Brokers buy/produce large chunks of scrip
Customers buy information with scrip

Figure 1 – Millicent’s Scheme

1. The Broker: The Broker mediates between Vendors and Customers in order to simplify the tasks they perform. He acts like a bank and provides the electronic currency (“scrip”) for the micro-payments. A Broker, after coming to a deal with the Vendor, can either generate his own valid “Vendor-specific” scrip, or buy a large amount of scrip, from the Vendor, using a macro-payment system. The Broker is then selling the scrip to the customers via macro-payment transactions. As it is seen, Brokers are just credit intermediates that buy huge amounts of scrip from the Vendors and sell large amounts of scrip to the Customers. During Customer purchases (either from Broker or Vendor), no transactions between the Broker and the Vendors are taking place.

2. The Customer: The Customers buy scrip from the Brokers, using real money, via a macro-payment system. The amount should be sufficient to cover the transaction cost plus to produce financial gain for both the Broker and the Vendor (scrip is Vendor specific). The Customer can then use the scrip to perform micro-payment purchases. No transactions with real money are taking place in any given time between Customers and Vendors.
3. The Vendor: The Vendor is the “data bank”. He supplies customers with data, services or both. He accepts his specific scrip as the only method of payment. The scrip was either generated by him (the Vendor) or by a licensed broker. Of course some validation and authentication is necessary to ensure that no double spending will take place. After that the Vendor can transmit the requested data back to the Customer, using a given encryption algorithm for avoiding fraudulent use.

According to DEC, initial tests of the Millicent implementation in Japan produced the following results:

- 14000 pieces of scrip can be produced per second
- 8000 payments can be validated per second.
- 1000 Millicent requests per second can be received from the network and validated.

- METEORE 2000: METEORE is a unified Internet/mobile payment solution for contents and services, to be used in the so-called "Mobility Portals". A mobility portal is defined as Web/WAP information based system, which provides information or services related to mobility:
    - Information's related to a geographical position (which can be the position of the consumer or the one specified by him) or movement (how to go from a point to another one)
    - Services like ticketing (entertainment, reservation, parking, etc.)
    - Emergency services: reception of SMS signalling events (strikes or delays for travels, stock exchange conditions, etc.)
    - Advertisement and advantages related to position or interest profile of the end-user.

  A mobility portal has the major characteristics to address multiple terminals: fixed terminals like PC's or mobile terminals like mobile phones or PDA's. It also addresses multiple payment modes: aggregated and single payment. The following figure[1] illustrates the high level architecture of the METEORE system.

[1] Figure 2 is owned by the METEORE consortium, which developed the METEORE system. The author was member of the consortium

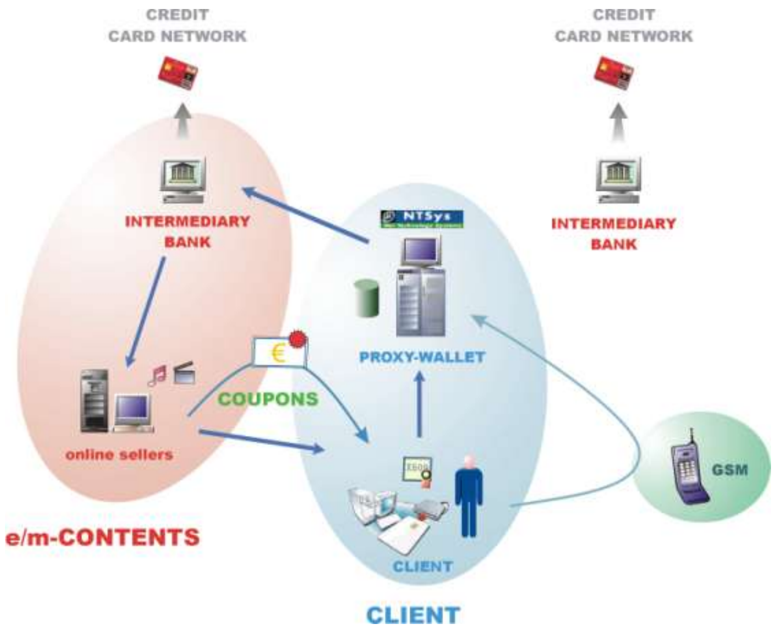

Figure 2 – METEORE's' Scheme

The client can access the sites of on-line sellers to buy coupons, which are stored in the CPS. The client can then buy e/m contents using these coupons, which the CPS authenticate with an intermediary bank. Alternatively the client can pre-pay the bank and create an account with Micro-COMM. The client can then use the CPS to buy e/m contents from online sellers, without dealing with the bank at all. The core system architecture combines an authentication layer at the core payment system that connects to an aggregation engine and a single payment gateway that interfaces to an external payment system in charge of authorisation and money transfers. Other important functional blocks are:

- Web back-offices: merchant back-office, consumer front-office, system/application back-office, that are all implemented as https portals,
- The system interconnection block.

The Core Payment System offers both aggregated and single payment mode, the authentication depending from the terminal capability. The METEORE model does not specify how the back-offices and front-offices work but only state their existence. Each implementation will use its specific interfaces.

Proxywallet is a payment access solution that interfaces to multiple banking systems and especially SSL bank intermediaries. It can be used as a unique access point either for direct connections to central authorisation/payment systems or to secondary access system like SSL intermediaries. ProxyWallet is used for the single payments.

Micro-COMM is a typical third party aggregation system built for contents. It uses strong authentication of users accessing via PC to contents sites, through a security agent that wraps communication on http. Micro-COMM is used for the aggregated payments.

Ease of Use

This section examines the actions/steps a customer has to undertake in order to use the system, and how complex these steps are. The section was summarized in table 1 (Source: rpcp.mit.edu/~pearah/micropayments/Survey.html).

	Relationships	**Model**
CyberCoin	Customer needs to establish a relationship with a bank that supports the CyberCash Wallet . Merchant needs to establish relationship with the CyberCash system	Debit card transactions; The pre-authorization of funds allows cost advantages due to less communication
Ecash	Customer needs to establish a relationship with a bank that supports the Digicash system. Vendor needs to establish relationship with the Digicash system	Withdrawing cash from an ATM By transferring the digital money into the HDD of the consumer, communication with the bank is avoided at the time of a transaction The vendor authenticates the digital cash during the transaction and then later transmits it to the issuer to check for double spending
NetBill	A NetBill server maintains accounts for both Clients and Vendors Clients replenish funds in their NetBill account as necessary using a credit card or bank account	Client requests product Vendor sends an encrypted version After verification of transfer takes place, Vendor solicits Broker a funds transfer Broker makes transfer and responds to Vendor Vendor forwards decryption key to Client
Millicent	Client enter into long-term relationships with brokers Brokers buy and sell vendor scrip as a service to Clients and vendors Once a Client’s scrip is set at the vendor, transactions take place and the broker only acts as accountant keeping record and managing scrip Consumer purchase scrip from brokers using a bank account or a credit card.	The client makes a secure connection to the broker to get some broker scrip If the client doesn't already have scrip for a particular vendor, he contacts the broker to buy some using his broker scrip If the broker doesn't already have scrip for that vendor, he buys some from the vendor The broker returns vendor scrip and change (in broker scrip) to the client The customer uses the vendor scrip to make a purchase from the vendor.
METEORE 2000	Client has to register with a bank that is using the system. Client has to register with the system. Client has to buy vendor specific coupons. Vendors have to register with system and bank.	Option for aggregated and single payment. Low cost contents can be bought using aggregated payments which minimise overheads, and normal cost contents can be bought using single payment. Setting up the system is easy and only requires an internet connection. Registering for using the system though is a complex and long-winded procedure. Once set-up, the system is easy to use and all complex procedures are hidden from the client.

Table 1 – MPS Ease of Use

Anonymity & Privacy

According to EU directives, micro-payment systems have to provide amongst other things anonymity and privacy for every single transaction. Both are not optional and are considered to be two of the most important features of any system making transactions with (S. Garfinkel '97). Table 2 summarizes how the five systems fulfill the above two requirements.

	Anonymity	**Privacy**
CyberCoin	No	Yes
Ecash	Yes	Yes
NetBill	Yes	Yes

METEORE 2000	Yes	Yes
Millicent	Depends on Millicent protocol (see bellow)	

Table 2 – MPS Anonymity & Privacy

- CyberCoin: The Client always has a record of his or her transactions and can prove them, but the Vendor will not know the Client's identity unless the Client reveals it.

- ECash: unlike paper cash, is unconditionally untraceable. The computations carried out by the Client's PC makes it impossible for anyone to link the payment to payer. Clients can prove that they did or did not make a payment, without revealing anything more. This level of privacy limits exposure to future data-privacy legislation and reduces record-keeping costs.

- NetBill: Client and account details are not shared with anyone except as required by law. Vendors are not provided with access to Client's proprietary information included in an electronic payment order. By design, the Vendor will never have access to the Client's bankcard number.

- METEORE 2000: The client does not reveal any personal details other that his/her digital signature, or phone number (that is registered with the system), and a password. The unique identifier of the client is his/her phone number.

- *Millicent: three distinct protocols enable different levels of privacy (and security) according to customer's needs. Table 3 (Source: (M. Pierce '97)) compares their characteristics.*
-

Protocol	Efficiency Ranking	Secure	Private
Scrip in the clear	1	No	No
Encrypted connection	3	Yes	Yes
Request signatures	2	Yes	No

Table 3 – Millicent's Security Protocols

  - *Scrip in the clear:* No network security is provided. The scrip is not getting encrypted. The purchased data are not getting encrypted either.
  - *Encrypted connection:* The scrip is getting encrypted using a symmetric algorithm. The encryption key uniquely identifies the Customer and the transaction. Data are getting encrypted as well.
  - *Request signatures:* Digital signatures are being used for uniquely identifying the transaction and the customer. Faster than the previous protocol because no encryption is taking place.

Security

CyberCoin: the financial information is encrypted and digitally signed, but the message itself is not.

eCash: provides the highest security possible by applying public key digital signature techniques. Additional security features include the protection of eCash withdrawals from the Client's account with a password that is only known to him; not even to his bank.

NetBill: uses a combination of public-key cryptography and a variant of symmetric-key cryptography to make sure that all its communications are secure, and all transactions are authorized. Their approach is based on the well-tested Kerberos protocol (Curry '92).

Millicent: Each transaction requires that the customer know the secret associated with the scrip. The protocol never sends the secret in the clear, so there is no risk due to eavesdropping. No piece of scrip can be reused, so a replay attack will fail. Each request is signed with the secret, so there is no way to intercept scrip and use the scrip to make a different request.

METEORE 2000: All transactions are done in XML, are digitally signed using a PKI, and are using an encrypted (128bit key) SSL carrier. The network itself is secured with firewalls and NIDSs, following a ”no-man” architecture. The administration is done remotely using SSL and X509 certificates, and the integrity is achieved using mirroring and backup techniques.

Micro-Payment Transactions

The broker is the party that provides and/or holds the electronic currency. The vendor is the party that provides the e/m contents. The following general transaction categories are taking place in the examined MPS:

1. Broker & Vendor. Must trust each other. The Vendor should provide what the Broker has promised to the Client. The Broker must pay the Vendor for the used scrip[2]. Both parties must have the same security procedures and patterns, as a chain is as secure as its weakest link.
2. Client & Broker. The Broker must comply with standards as he is holding personal information. Data security is of the essence as the client trust is what keeps Brokers in business. The transactions between these two players are macro-payment transactions. Broker has to apply relevant security solutions to the following problems: Privacy infringement, Authentication, Repudiation, Integrity, Denial of service, The Broker must provide the Client with valid scrip.
3. Client & Vendor. The Vendor must be able to authenticate and validate the scrip. The Vendor must always be able to provide data (denial of service, interception problem).

As with all systems we thrive for confidentiality, integrity availability, and non-repudiation. These concepts constitute three of the four goals of information security, and are examined later in this chapter.

1. Confidentiality: Assets must be accessible by authorized parties, hence:
   - Client's' private data must remain private in Broker's' database (database security, transaction security).
   - Client's' scrip should not be “stolen” by third parties (transaction security, encryption issues).

[2] Scrip: electronic currency or other feature that is used as electronic currency from the MPS

- Vendor's' data are accessible by valid Clients only (authentication, repudiation).

2. Integrity: Assets can be modified by authorized parties and in authorized ways, hence:
    - Brokers must issue valid scrip (trust, forgery issues).
    - Scrip can only be generated by valid Brokers (authentication, repudiation).
    - Vendor's data cannot be modified by third parties (database security, transaction security, encryption issues).
3. Availability: Assets must always be accessible to authorized parties, hence:
    - Broker must always be able to generate and supply scrip to clients (denial of service issues, software & hardware protection issues)
    - Vendor must always be able to supply data to valid clients (denial of service issues, software & hardware protection issues)

The above security requirements are in agreement with the logical security layer of on-line transactions that was presented by Hoogenboom and Steemers (Hoogenboom '00). According to them:

- Both parties must be able to authenticate each other (mutual authentication).
- The integrity of the information exchanges must be verifiable (data integrity).
- The ordering party must have access to the desired service (service access).
- The confidentiality of the transaction must be guaranteed (information encryption).
- The transaction must be verified through at least two channels.

Micro-Payment Systems Vulnerabilities

As with all computing systems, micro-payment systems have vulnerabilities that can be categorised by the following broad categories of their resources:

- Hardware
- Software
- Data
- Administrative
- Physical

This section contains vulnerabilities that were observed during the examination of the various MPS. No tool was used to identify them other than the knowledge of the author.

Hardware

All hardware components that are not necessary for the operation of the MPS should be removed. By hardware we mean floppy drives, CD-ROMs, parallel and serial ports as well as USB and infrared ports. Furthermore the unused data ports on the motherboard itself should be disabled or removed. The option of the wireless network is not an option as the security risk is so high that the cost involved for securing such a system does not justify it.

All computers that participate in the MPS should be protected by a BIOS password. A member of staff though with physical access to the circuits of those computers can bypass this security feature by removing the battery of the motherboard for an approximate of 1 minute (as tested in our labs). Anything less and the voltage on the circuit itself will be able to maintain the CMOS data. The test in our lab showed that a hostile internal attacker (dissatisfied member of staff, member of staff employed by a competitor…) with a screwdriver as his only weapon only needs an approximate of 3 minutes to bypass the BIOS password.

After gaining access to the BIOS an attacker can choose to boot the computer from an alternative device, hence it is essential to have the absolute minimum hardware installed. He will even be able to install an alternative HDD and use it to retrieve data that otherwise it would be impossible for him to do. After an experiment in our lab, we where able to retrieve the private data of a Debian user, as well as all the sensitive configuration files of the operating system. The suggested countermeasure to the above vulnerability is to physically secure the room that the computers are in and physically lock their cases to the ground. Furthermore, a sophisticated lock should be installed between the cases of the computers and their covers to prevent access to the computer circuits by the use of a simple screwdriver.

There have been a lot of discussions (Barber '01), (Fraser '97), (K. Pagan '97), (Pounder '01), (Scambray '01), (Stalling '00), (Sutton '99) on network security and how to better secure a corporate network. One of the common outcomes is the single point of entry. Should there is only one path for an attacker to come in the CPS then the ISec officers will have an easier job to do. That was not the case though to some of the MPS that were examined. Every single point of entry should have the same level of security. The weakest link destroys the game altogether. The two concepts that the ISec officers should be concerned with is the robustness of each machine participating in the MPS and the robustness of the connection between the servers and between the servers and the clients.

Software

Banners

All services and applications in general, identify themselves by giving away into the world "banners". The "banners" contain all the necessary information for an individual or a software program to recognise exactly the type of service, its version and some times set-up parameters as well. As it was discussed in other sections of this report, threat agents can use these "banners" to collect information for the system and then interrogate vulnerability databases in order to find ways to break them. Because of the nature of the systems it is best practice to put fraudulent information in those "banners" in order to mislead any potential attackers.

Services

According to (Barber '01), (Carroll '96), (Fraser '97), (Stalling '00) only the necessary services should be offered from each computer that is part of any computing system. Even if a computer is not directly connected to the Internet it has to be certain that no unnecessary networking services are offered. You can never have too many security layers installed in a system. Only secure and authorised computers should be part of the system. Firewalls (K. Pagan '97), (Guruz '01) must be installed between the intranets and the Internet, and also between the Intranets themselves. The firewalls must not only check the sessions but the packets as well. All the firewalls must offer the same level of security and protection. A port-scanner should be used in a daily basis to monitor for unauthorised services.

Network Traffic

Cryptographic techniques (M. Hoogenboom '00), (Write '96), (Schneier '96) should be used for all network traffic, both inside and outside the system. The financial and personal data of the users, stored in the MPS, should also be encrypted. Such a countermeasure will prevent the modification of the data in the case of an intruder gaining access to the servers. The MPS should be a no-man zone. No user should be allowed to run any sort of programs on the network other than the ones dedicated to maintenance.

DNS

Some MPS are using the DNS service to simplify their operations. The question is what computers are contained in the DNS database and which computers are running the service. The DNS should not be public in order to minimize the possibility of an attacker gaining information about the private machines running the MPS. Furthermore, there should be a backup DNS for robustness purposes. Will that server be allowed to perform zone transfers? If yes, this poses a major vulnerability that needs to be countered at any costs. Zone transfers must be restricted to authorized servers only. The "allow-transfer" directive in the "named.conf" file must be used to enforce the above restriction. Furthermore, the firewall must be configured to deny all inbound connections to TCP port 53.

One common implementation of the DNS is BIND, which was used by most MPS implementations. The first step in setting up BIND is the creation of the DNS data. The configuration details are spread over multiple files. For reference purposes the file that maps host names to addresses will be called db.wallet and the file that maps addresses to host names will be called db.addr. Two more configuration files are the db.cache and the db.127.0.0. To tie all the files together a startup file is required. The file is normally called named.boot and it is stored in the /etc directory. Needless to say that the permissions for this file should be the same as for the /etc directory: -rw-r- -r- -. For maintenance purposes at least one user should have write permissions, but his actions should be audited, as an unauthorized entry to this file, even if unintentional, will cause a denial of service.

It is a norm for the entries in the db files to have the following structure: SOA record, NS record, Other records. The SOA stands for start of authority and indicates the best source of information for a zone. There can be only one such entry, as there can be only one such machine in each zone. The NS stands for name server and contains the records for the name servers of the zone. HINFO records must NOT be used in the DNS database as they contain very sensitive information that does not offer any sort of functionality other that making DNS administration easier.

One of the most popular attacks against domain name servers is the zone transfer. A legitimate zone transfer allows a secondary master server to update its configuration files from the primary master DNS. Only a secondary master DNS should be able to issue a zone transfer request. In our system then, only the secondary firewall must be authorized to interrogate the primary one and ask for a zone transfer.

The simplest way to perform a zone transfer is to use nslookup. By setting nslookup to interrogate the primary DNS of the MPS, we can issue the following commands:

```
C:\Documemts and Settings\user>nslookup
Default Server: mydns.glam.ac.uk
Address: 10.10.10.51
>set type=any
>ls –d MPS.net. >> /tmp/MPS_zone
>
```

The file /tmp/MPS_zone in the local machine of the attacker, will now contain information about all the computers that are part of the MPS.

As mentioned before, only authorized servers should be able to issue zone transfer requests. This can be achieved by using the xfernets directive in the named.root file. On the network side, the firewall could be set-up to deny all unauthorized connections to TCP port 53 (the zone transfer requests are TCP:53).

IDS

There is a variety of IDSs used by the examined MPSs. SNORT (Roesch '01) was the most popular. SNORT is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks (Roesch '01). It can produce documentation “like there is no tomorrow”. Because the MPS is handling financial transactions, we believe that legal advice is required to ensure that the output of SNORT will be able to stand in a court of law.

SNORT should log all traffic to and from the open ports. The best way of stopping a future attack is by identifying it when it is on the “information gathering” stage. By closely monitoring ICMP and nslookup requests, and by detecting portscans with the relevant SNORT preprocessor, attackers can be identified and legal actions can be taken. Another preprocessor that should be used is the “Frag2”. “Frag2” allows SNORT to perform a full-blown IP de-fragmentation, making it more difficult for hackers to circumvent the detection capabilities of the system. For TCP stream monitoring the “Stream4” preprocessor should be used. TCP streams on the configured ports with small segments will be reassembled and properly evaluated for malicious activity. “Stream4” is able to handle 64.000 simultaneous TCP connections.

To close this section, SNORT rules must be nested. Because there is a need for speed and efficiency, complicated and slow contents rules should only by called when something suspicious has been identified by using other faster connection rules. A new administrator should only require a couple of minutes to understand and follow the rules “out of the blue”. If the case is otherwise then there is a need to revise the rules and make them more efficient.

Firewalls

Most firewalls were based on the netfilter application. Netfilter is a packet-filtering gateway, which uses ipchains/iptables rules for effectively checking every single packet that comes and goes from the configured ports. First of all the firewalls should not reply to echo requests nor to any type of portscans. It has been observed that once an attacker will identify a firewall it will try to work around it and not through it. Hence, by making it difficult for an attacker to identify firewall machines, we earn more time for allowing the IDS system to successfully identify the attack. Another countermeasure that could be deployed is against route tracing. We do that by restricting firewalls from responding to TTL expired packets.

ICMP and UDP tunneling should also be monitored and trapped. ICMP tunneling is the capability of wrapping real data in an ICMP header. ICMP traffic should not be blindly allowed through the firewall. By limiting ICMP traffic through the firewall we also secure our system against a lot of denial of service attacks. By not allowing broadcast ICMP echo requests we effectively protect the system from “smurf” attacks.

Fragments must also be checked on the firewalls. Sanity checking on the fragmentation length must be performed (both for being too small or too large). Carefully constructed packets send to the firewall machines of the MPS would result on a system reboot or a system halt.

CGI Security

Some of the MPS were using CGIs for communication purposes. There are many ways to prevent CGI attacks and the easiest and most sensible is to install the web server correctly from the start. Vulnerabilities in the web program should be checked with Mitre.org, CERT (MITRE '02) and Bugtraq (Security_Focus '02) sites and the web server's manufacturer for patches and updates. The operating system itself must of course be brought up to date with the latest patches. All web CGI programs must be checked for insecurities via scanners to see if they are tempting targets for hackers.

According to (Jawi '01), (McKay '01) the developers must take under consideration the following rules:

- Always include error-handling code to warn if the file isn't actually a file, cannot be created or opened, already exists, doesn't exist, requires different permissions, and so on.
- Watch which directories are used to create or open files. Never write a file to a world-writeable or world-readable directory.
- Always explicitly set the file's UMASK.
- Set the file permissions as restrictively as possible. If the file is a dump of user input, such as a visitor list, the file should be readable only by the processes that will engage that file.
- Ensure that the file's name does not have metacharacters in it, and if the file is generated on the fly, include a screening process to weed out such characters.
- Never run the web servers with root privileges.
- Delete scripts that are not in use to prevent a vulnerability from being exploited.

Utilize CGIWrap written by Nathan Neulinger to allow general users to use CGI scripts and HTML forms without compromising the security of the http server. Scripts should not be using the user ID of the httpd process. Also, several security checks are performed by CGIWrap on the script, which will not be executed if any checks fail.

Network Reconnaissance Countermeasures

SNORT (Roesch '01), (D. Wreski '00), (Roesch '00) is one of the best available sources (both open and close) for detecting reconnaissance activity. The above tool can do nothing though for preventing a hostile party from gathering sensitive information from a system. According to (Scambray '01) the tool that should be used for that is called "RotoRouter"[3] and counters the above vulnerability by generating fake "traceroute" responses.

Spoofing Countermeasures

According to (Scambray '01), (Sutton '99) no two users must be allowed in the same computer in the same time. We must keep in mind that new vulnerabilities are surfacing every day. By not allowing the above we make sure that even if the computer suffer from uncovered vulnerabilities, no one will be able to exploit them using a spoofing attack.

Ping Sweeps Countermeasures

Ping sweeps are the first step of an active attack. Being able to detect them and find out their source will greatly help in efficiently protecting the network. The primary method for detecting ping sweep attacks is network based IDS programs such as SNORT. An example SNORT rule set can be seen in appendix A-2.

[3] http://onlinesecurity.virtualave.net/attacks/tools/roto.htm

According to (Sutton '99), (M. Hoogenboom '00), (Scambray '01), unless otherwise needed by the system, only the following types of ICMP packets should be allowed through the firewall:

- ICMP ECHO_REPLY
- ICMP HOST_UNREACHABLE
-

SYN Flooding Countermeasures

Since 1996 the SYN flood has made news with attacks against the FBI, the White House, NYSE web sites, along with many others. The use of random spoofed source IP numbers on the IP headers made tracing the attacks back to the source nearly impossible. Hackers developed ‘Zombie" hosts to attack targets in unison from a master machine. The technique, now famous, is referred to as the distributed denial of service attack. It uses many machines to coordinate attacks, and in many cases without the owner’s knowledge or consent. The result is the same. The target computer fills a very small queue of half open port connections. Once the queue (also called a Backlog) is full, no other connections can be made until a connection times out and the memory space cleared, or the connection is complemented, which moves the connection out of the queue into an application level memory buffer. The computer stops answering requests on the attacked port once the Backlog threshold is reached. In the case of web servers, the attacks are aimed at ports 80 or 443. Once overwhelmed, the Web server stops working as designed and the attack is successful.

The defense against a SYN flood DoS attack must start at an organizations boarder router and continue through the network all the way to the target host. Boarder routers of organizations usually represent the boundary between an organizations line of control and responsibility and the wild world. Boarder routers represent the first organizational network device a packet travels through on the way to the target host. A limit of half open connections that can be established through that router must be set. Once the limit is crossed, the router will begin to drop connections at a specified rate. The aggressive interception value is usually half of the normal network threshold of half connections.

The next level of defense comes form the firewall machines. Proxy firewalls are the best choice as they can detect the states of the connections and wait for them to be full before forwarding them to the actual hosts. The iptables firewall though is a packet firewall and cannot detect the states of the connections. A change is not recommended for the trial or for a small-scale implementation in a single geographical area. Should the consortium decides to go live with a large scale implementation of the system over a number of geographical areas, then an investigation for justifying the expense of a proxy firewall must be conducted.

SQL Injection Countermeasures

Most MPS are using some version of SQL to hold data. The following list present defenses against the SQL injection attack:

- Full XML source code audit
  - Review all SQL queries and all variables in them
  - Trace SQL queries and look for user data
- Take advantage of stored procedures
- Minimize privileges of database connections
- Disable verbose error messages.

Data Vulnerabilities

Key Management

According to (Schneier '96), (Wright '97), (Write '96), (Sutton '99):

- The system itself should manage all the sensitive cryptographic elements such as keys and algorithms.
- Customers and users should not be able to read or modify sensitive cryptographic elements.
- The system should not serve user and customer requests that are not secure and encrypted by the default ciphers with authorised keys.

The keys should be 128bit in length. “Bruteforce” attacks are common and the system should be able to effectively counter them. Calculating the complexity of a bruteforce attack is relatively easy. If the key is 56bits long there are 2^{56} possible keys. If we assume that a supercomputer can try a million keys per second, it will take 2285 years to find the correct key. There is always a 50% chance, after we have tried half of them, to accidentally find a match. If the key is 128 bits long, the same supercomputer would require 10^{25} years.

Key management is the hardest part of cryptography. A threat agent will not spend millions of Euros if he can spend a couple of thousand to bribe a user to get his keys. An initial question that arises is if the customers have knowledge of what their key is. If they do then the customers will be prone to social engineering attacks. In such a case, the lifetime of the key should be very short.

Another concern is that only active customers should have access to the system, and all the dormant accounts should be getting deleted in a regular basis.

Verifying Keys

The third issue is the key verification. The problem in the computing world has the name of: man-in-the-middle. This method of attack is discussed in another section of this report. Public key cryptography used with digital signatures and key distribution centres is considered to be the best solution. The capabilities required for substituting a key are over those of the category of hackers, crackers, and script-kiddies. Of course organised crime organisations, and rival companies have both the resources and the money to do so.

To counter this vulnerability the system could be using customer certificates. In order for the certificates to work though, we must ensure that they are not jeopardized. The only 100% safe way to achieve the above is to send the certificates to the customers using floppy disks via recorded delivery. Crime organizations though can influence almost everything. An even safer scenario would be the use of the branches of the bank that the customer is registered. Most MPS actively involve one or more banking organizations. Because the system is dealing with electronic money the fact that the customer should have to go and collect something from his bank should not be a drawback. More of the point, customers are used in collecting their credit cards and their pin numbers form their banks so they will view it as another safety trigger against possible misuse.

Password Management

According to the NSA guidelines (Sutton '99) the first password for the user of the system and the first PIN for the customer should be temporary, in order to give them sufficient access to the system for changing them. The system must automatically ensure that the change must happen in a fixed time period after the receipt of the first password or PIN. If that period expires before the connection of the customer, the customer should only have the option of a new temporary PIN (for a fixed period of time).

According to (Ongetta '99) the system must keep a history of passwords and PINs and prevent the reuse of the same ones. On top of that, the system must fix the minimum life of the password and PIN as well as the maximum to prevent users changing their passwords until they can reuse their favourite one. The minimum length of the password and PIN for the users and the customers should be eight characters.

The administrators should not use their login name in any form, as a password. Furthermore they should not use their first or second name or even any nicknames that they might have. Passwords should not be words contained in English or other dictionaries, spelling lists or other lists of words. Passwords should not be based on information that are easily obtained from electronic or other sources. Passwords should be a combination of alphanumeric and punctuation characters

To sum up this section, the following three conditions must be fixed:

- A limit to the number of incorrect password and PIN attempts. We suggest that to be set to 0.
- The minimum length of the password/PIN and the type of characters used. We suggest a minimum of 8.
- The lifetime of the password.

Ongetta has compiled a best practice on putting together a password policy (Ongetta '99).

Log Files

All actions should be monitored, especially on the servers of the MPS. It is important to keep all log data in such a way that they will be able to stand in a court of law. As it was discussed, user trust is essential for the survivability of an MPS. That means that in the possibility of a breach, the system will be able to trace it back, find the responsible entity and legally prosecute it.

According to (Sutton '99), (Fraser '97), (Scambray '01) only authenticated users should be able to view the log files and perform authenticated actions. Furthermore, no one should have permissions to amend the log files in any way. The log files should be baked up daily, and the back-ups should be treated in the same way as the log files.

Because different stakeholders host different parts of the system there is a need for the security team to be able to cross-reference all of the log files. A tool should be developed in order to effectively solve the issue of data fusion and data mining. This issue must be addressed in the information security policy document.

Administrative Vulnerabilities

Information Security Policy Document

A policy document should be approved by the stakeholders of the MPS, published and communicated to all the partners accordingly. According to ISO17799 (BSI '00) the following guidance should be included as a minimum:

- *A definition of information security*
- *A statement of management intent*
- *Explanation of the security policies that are important to the organization, for example:*
  - *Compliance with legislative and contractual requirements*
  - *Security education requirements*
  - *Prevention and detection of malicious software*
  - *Consequences of security policy violations*
  - *References to supporting documentation*

Having a number of alarms and countermeasures deployed in the system for ensuring its safe operation is considered to be best practice. The alarms though can only inform about malicious actions against the system, they cannot prevent them. It is important for the administrators of the system to know exactly the procedures that they will have to follow for each case of an alarm switching on. As Schneier (Schneier '01) is saying: "Real security is about people."

All the alarms and countermeasures should be specified and analysed in the security policy document. Because different stakeholders are involved in an MPS, the security policy document should comply with the security policy document of each one of them.

The security policy document should take under consideration at least the following two security standards: ISO17799 (BSI '00) & ISO15408 (ISO/IEC '98). According to the number of European countries the system will be deployed different laws and legislations will probably apply. Legal advice for each country is required.

The security policy document should not be made publicly available, as it will contain sensitive information about the security procedures and countermeasures of the METEORE system. Instead a second document containing the management views and support as well as a small overview of the whole document should be made ready for any customer requests. According to our research the major asset, and major vulnerability in the same time, of such systems is the user's trust (see Kanani (Kanani '03). The users must know that there is a policy document that covers all the possible security concerns the system will have to face during its operation.

The employees that will develop the security policy document must be internal full time employees bound by non-disclosure agreements and not external contractors.

According to the NSA guidelines (Sutton '99), information security is a business responsibility shared by all members of the management team. There is a need to identify the security responsibilities of each one of them and include them in the information security policy document. The above is compliant with the two security standards we are concerned with: the ISO17799 (BSI '00) and the ISO15408 (ISO/IEC '98).

A cross-functional security team should be assembled. Such team will be responsible for:

- Agreeing specific roles and responsibilities for the different stakeholders,
- Agreeing on the information security methodologies,
- Agreeing and supporting information security initiatives
- Coordinating the implementation of specific information security measures.

The Root Account

All computers have a local administrator account. Its default name under UNIX is root, and under windows is administrator. There are a lot of “brute force” programs (Barber '01), (Scambray '01) in the public domain that are exploiting the above vulnerability. By renaming the root account, the job of such programs becomes a lot more difficult as they would have to “brute force” both the account name and the password field.

According to the NSA guidelines (Sutton '99) the “root” account should not be used for everyday tasks. It should only be used for maintenance and as a last resort. There should be dedicated accounts for running specific tasks with each account having specific privileges. The password of the “root” account should not be known to the administrators other wise they would be tempted to use it. Rather a hard copy of the password should be safely stored in a secure location and only a member of the managerial staff should have access to it.

The root password should not be the same across all the computers of the system. The vital servers that are running important services should each have a different root password. This prevent an attacker from a deep penetration should he acquires a root password of a less important, hence less protected, computer. A user should not be allowed to remotely connect to a computer using the root account of that computer. All passwords should be changed every time a member of the company that had administrative privileges leaves the company. The threat of a dissatisfied employee actively attacking the company is real and according to some statistics very high as well.

User & Customer Responsibilities

According to NSA’s security guidelines (Sutton '99):

- Users and customers must faithfully follow the security policies, especially the ones for choosing PINs and passwords.
- Most users will right down their passwords no matter the policy. It would be better to develop a policy that would address the above.
- User’s system password should not be the same as any other passwords
- Passwords should not be stored in files in the file system of their computer
- Users must be careful when they are moving or copying files. The file permission should be thoroughly checked.
- Users should not be allowed to install or run any sort of software programs in the system. It is believed that there are a lot of Trojan horses on the Internet that have not yet been discovered.

Because of the financial nature of the system it is important to have strong personnel screening procedures and policies. Because customer trust is the biggest asset and vulnerability of such a system, it is essential that these procedures and policies are publicly available. All employees working with the MPS under all the different partners must sign confidentiality or non-disclosure agreements.

Internet Footprint

Information about the company and the payment system will be publicly available. By information we mean:

- locations
- related companies & entities
- phone numbers
- contact names & e-mail addresses
- privacy or security policies indicating the types of security mechanisms in place
- links to other servers related to the organisation and the payment system

It is very important to evaluate and classify the type of information that is publicly disseminated. Without that information an outsider will not have the capability to deploy an active attack against the system. Instructions on how to counter the above vulnerability can be found at (Fraser '97). By effectively addressing that vulnerability we limit the source of the potential threats against the system to those coming from the “insiders”.

This very important issue must be clearly discussed in the information security policy. A good policy will by effectively minimize the threat of the “script-kiddies”.

Physical Vulnerabilities

According to ISO17799 (BSI '00) the following guidelines should be implemented:

- The security perimeter should be clearly defined
- The perimeter of the building where the server room is situated should be physically sound. Furthermore all the external doors should be suitably protected against unauthorised access.
- The physical access to the building and the room should be controlled by sufficient mechanisms that will not be allowing unauthorised access.
- Physical barriers able to isolate sections of the building and the room should be in place. The barriers must be able to withstand environmental contamination.

The above are countermeasures against hostile external threat agents, and against hostile internal ones without access to the secure room. There is no point in having only one stakeholder deploying countermeasures against physical vulnerabilities. A threat agent with the proper motivation and capabilities will always perform the intelligence-gathering phase, hence will always be able to identify the easiest way to bypass the system defenses.

References