JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

SYT Matura

Ausarbeitung

Systemtechnik
5BHITT 2015/16

Niklas Hohenwarter

Version 1.0

Inhaltsverzeichnis

Kompetenzen

Dezentrale Systeme

1 Können die in dokumentenbasierten, nachrichtenorientierten und serviceorientierten Systemen eingesetzten offenen Dokumentenformate und Auszeichnungssprachen erläutern
2 Können Sicherheitskonzepte für verteilte, dezentrale Systeme entwickeln
3 Können ausfallsichere replizierte Datenbanksysteme und dezentrale Systeme installieren, warten und entwerfen
4 Können verteilte und redundante Dateisysteme einsetzen

Systemintegration und Infrastruktur

1 Können in Unternehmensnetzwerken ausfallsichere und redundante informationstechnische Systemarchitekturen mit unterschiedlichen Betriebssystemen realisieren
2 Können Sicherheitskonzepte für die unternehmensinterne und unternehmensübergreifende Kommunikation umsetzen
3 Können Lastverteilung auf Netzwerkebene realisieren
4 Können Fernwartungstechniken beschreiben und diese im Unternehmen geeignet einsetzen

1 Cloud Computing und Internet of Things

1.1 IoT Datenübertragung

Eine der wichtigsten Entscheidungen bei der Übertragung von Daten ist die Wahl des Datenformates. Heutzutage sind die Datenformate XML und JSON am meisten verbreitet. Jedes dieser Formate hat einige Vor- und Nachteile [1]:

JSON

+ Einfache „kleinere" Syntax und damit weniger Overhead
+ Gute Kompatibilität mit JavaScript und somit sehr geeignet für die Webentwicklung
+ JSON Schema kann zur Validierung verwendet werden
+ JsonPath um Daten in komplizierten JSON Dokumenten einfach zu finden
– Wenige Datentypen

XML

+ Allgemeines Format; Es können eigene Dialekte erfunden werden
+ XML Schema zur Validierung und Erstellung neuer Datentypen
+ XSLT um XML in andere Formate Umwandeln zu können (z.B. HTML)
+ XPath/XQuery zum Finden von Daten in komplizierten Dokumenten
+ Support für Namespaces
– Großer Overhead

Einige sind jedoch der Meinung, dass man XML und JSON nicht direkt vergleichen kann. JSON wurde dafür entwickelt, Daten mit wenig Overhead übertragen zu können. XML kann hier jedoch viel mehr und wird bei komplexeren Datenstrukturen und wenn Validierung des Dokumentes wichtig ist heute noch häufig Eingesetzt. Aus XML haben sich auch viele andere Sprachen wie z.B. SVG und WSDL entwickelt [2].

1.2 IoT Systemarchitektur bei Cloud Anbieter

Es bietet sich an eine IoT Verwaltungsplattform bei einem Cloud-Anbieter zu hosten. Dadurch kann diese schnell skaliert werden und auch einfach neue Hosting-Standorte hinzugefügt werden.

Alle bekannten Cloud Anbieter bieten einen Loadbalancing Service. Dieser sollte verwendet werden, da damit einfacher skaliert werden kann. Um die Services der Plattform aufrecht zu erhalten, sollten diese in Containern mit einem Container Management wie z.B. Docker deployed werden. Dann kann mit dem Management Tool Kubernetes dafür gesorgt werden, dass z.B. immer drei Webserver zur Verfügung stehen. Diese Webserver bekommen dann ihre Anfragen über den Loadbalancer des Cloud-Anbieters. Bei Cloud Anbietern

kann allerdings auch ohne Kubernetes gut skaliert werden. Es gibt die Möglichkeit nach Auslastung der VMs einfach automatisch neue erstellen zu lassen [3].

Zum heutigen Zeitpunkt gibt es drei große Cloud Anbieter: Amazon AWS, Google Cloud und Microsoft Azure. AWS ist schon seit 2012 auf dem Markt, die anderen beiden Anbieter gibt es erst seit Mitte 2013. Anfangs bat AWS mehr Funktionen, jedoch glich sich das über Zeit aus. Aktuell sind die gebotenen Features relativ ähnlich und haben Großteils nur andere Namen. Google scheint aktuell jedoch am billigsten zu sein und am stärksten zu wachsen [4].

1.3 IoT Kommunikation

Es gibt drei mögliche Arten um ein IoT Netwerk aufzubauen: Stern Topologie, Peer-to-Peer und Hub and Spoke Topologie. Jede dieser Topologien hat Vor- und Nachteile[5]:

Stern Topologie: In einer Stern Topologie kommuniziert jedes IoT Gerät mit einer zentralen Stelle – z.B. ein Cloud Service. Dieses Prinzip wird heutzutage Großteils im Internet verwendet – ein Server mit welchem viele Clients Informationen austauschen. Der Vorteil dieser Topologie ist, dass es eine zentrale Verwaltung gibt. Die Daten werden zentral beim Cloud Dienst gespeichert und verarbeitet. Die einzelnen Geräte müssen nur wissen, wie sie Kontakt zur „Zentrale" aufnehmen. Der Nachteil hierbei ist, dass alle Daten an einer Stelle gespeichert und verarbeitet werden. Diese ist dann Anfällig für Angriffe und Ausfälle. Außerdem wird es bei steigender Useranzahl komplizierter zu skalieren und die Latenz steigt an. [5]

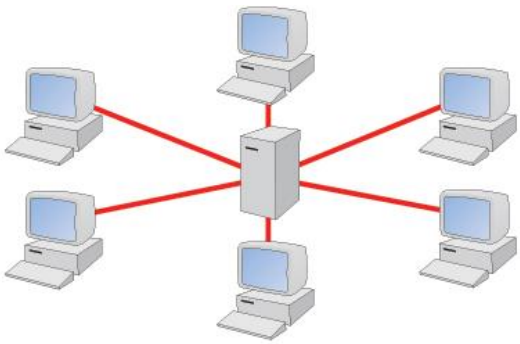

Abbildung 1: Stern Topologie [6]

Peer-to-Peer: In dieser Topologie kommunizieren alle Geräte direkt miteinander und es gibt keine zentrale Anlaufstelle. Der Vorteil hierbei ist, dass das Scaling sehr einfach ist, da es ja keine zentrale Belastung gibt und die Last auf alle Geräte aufgeteilt ist. Sicherheitstechnisch ist diese Topologie ebenfalls besser, da jetzt mehrere Geräte gehackt werden müssen um große Mengen von Daten zu erhalten. Die Latenz ist ebenfalls niedriger. Der Nachteil von P2P ist, dass die Verwaltung um einiges komplizierter ist. Die Daten werden nicht an einem Punkt gespeichert und sind somit schwieriger zu analysieren. [5]

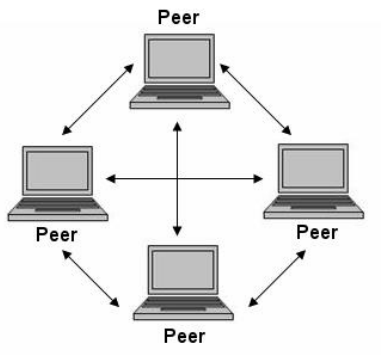

Abbildung 2: Peer to Peer [7]

Hub and Spoke: Diese Topologie ist ein Mittelweg zwischen den beiden zuvor genannten. Hier kommunizieren z.B. alle IoT Geräte eines Haushaltes mit einem lokalen Hub. Dieser Hub verarbeitet die Daten teilweise gleich lokal. Andere Daten werden an eine zentrale Plattform weitergeleitet. Diese Topologie bietet die gleichen Vorteile wie P2P und ein relativ einfaches Management. Jedoch gibt es nun auch wieder einen Single Point of Failure im System, nämlich den Hub. Wenn dieser Ausfällt, können keine Daten mehr verarbeitet oder weitergeleitet werden. Außerdem gibt es aktuell bei solchen Systemen für Smart Homes große Unterschiede bei den Herstellern und die Sensoren eines anderen Herstellers sind oft nicht mit dem Hub kompatibel. [5]

Abbildung 3: Hub and Spoke [8]

1.4 IoT Sensorsystem Fernwartung

Das Sensorsystem kann z.B. über REST die gemessenen Werte an einen Server übertragen. Die Werte werden dann in einer Datenbank abgelegt und auf der Plattform angezeigt. Die einzelnen Sensoren im Netzwerk könnten mithilfe von DNS identifiziert werden. Eine gute Identifizierung wäre im Falle des Bojen Beispiels <ID>.<SEA_NAME>.bojen.com. Somit könnte eine bestimmte Boje einfach identifiziert werden. In einer zentralen Datenbank kann dann die Adresse der Boje zusammen mit ihren Koordinaten abgespeichert werden. [9]

Im Falle von defekten oder falschen Werten kann über die DNS Adresse auf eine IP Adresse aufgelöst werden. Nun muss nur noch auf das System der Boje zugegriffen werden. Zur Fernwartung steht eine Vielzahl von Protokollen zur Verfügung. Am häufigsten werden SSH, RDP und VNC verwendet um auf entfernte Geräte zuzugreifen. Im Falle von IoT Geräten kommt jedoch eigentlich nur SSH in Frage, da die anderen Beiden Protokolle zur Übertragung von Desktop Umgebungen entwickelt wurden. Somit bleibt nur noch SSH zur sicheren Fernwartung über. Um eine SSH Verbindung möglichst gut abzusichern sollten keine Passwörter sondern RSA Keys verwendet werden. Über SSH kann dann auf das IoT Gerät zugegriffen und dieses gewartet werden. [10]

1.5 Verteilte Dateisysteme

Bei verteilten Dateisystemen kann auf diese zugegriffen werden als wären sie Lokal. Dadurch ist es möglich z.B. in Clustern dedizierte Rechen-Nodes und dedizierte Speicher-Nodes zu haben. Auf die Dateisysteme wird dann übers Netzwerk zugegriffen und die Dateien lokal gecacht. Drei bekannte DFS(Distributed File System) sind OCFS2, HDFS und GFS. [11]

OCFS2: Dieses Filesystem kann dazu verwendet werden um auf Shared Storage wie iSCSI zuzugreifen. Beispielsweise kann eine VM mittles OCFS2 ein Volume einbinden. Auf dieser eingebundenen Festplatte liegen dann die Produktivdaten, welche persistent gehalten werden müssen. Somit können beliebig viele VMs zu dieser Festplatte verbunden werden und auch nach Belieben gelöscht werden. [12]

HDFS: Das Ziel des Hadoop File Systems war es besonders Fehlertolerant zu sein. Dies ist besonders wichtig, wenn Server mit Consumer Grade Hardware verwendet werden, da diese eine höhere Ausfallwahrscheinlichkeit und Häufigkeit haben. Die Daten werden in Blöcken mit einer Standardgröße von 64MB gespeichert. Das System arbeitet nach dem Master-Slave Prinzip. Die Daten werden redundant gespeichert. Ein HDFS hat mindestens einen Name Node welcher die Metatdata-Operationen übernimmt. Des Weiteren gibt es n Data Nodes welche die Daten speichern und die Block-Operationen durchführen. [13]

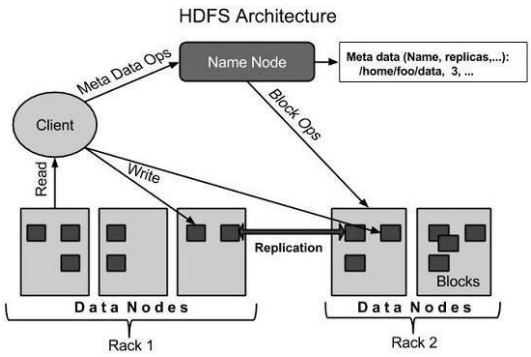

Abbildung 4: HDFS Struktur [13]

GFS: Das Google File System wurde von Google für ihre Server entwickelt und ist nicht öffentlich zugänglich. Das HDFS wurde anhand des GFS entwickelt. GFS besteht aus einem Master Server und n Chunk Servern. Die Daten werden in Chunks gespeichert. Ein Chunk hat eine Größe von 64MB. Der Master Server teilt jedem Chunk eine eindeutige 64bit ID zu. Darüber können die Chunks nachher wieder gefunden werden. Die einzelnen Chunks werden auf den Chunk Servern als Dateien gespeichert. Normalerweise gibt es von jedem Chunk drei Replikate um die Verfügbarkeit der Daten zu gewährleisten. Der Master Server speichert alle Metadaten. Dazu gehören Zugriffsberechtigungen, das Mapping von Dateien auf Chunks und den Chunkserver auf welchem ein Chunk liegt. Applikationen welche auf das Dateisystem zugreifen wollen müssen die GFS API implementieren. Wenn zugegriffen wird, kommuniziert die Applikation mit dem Master Server um Metadaten zu lesen/schreiben. Die eigentlichen Schreibe- oder Leseoperationen laufen nicht über den Master Server sondern finden direkt zwischen Client und Chunk Server statt. [14]

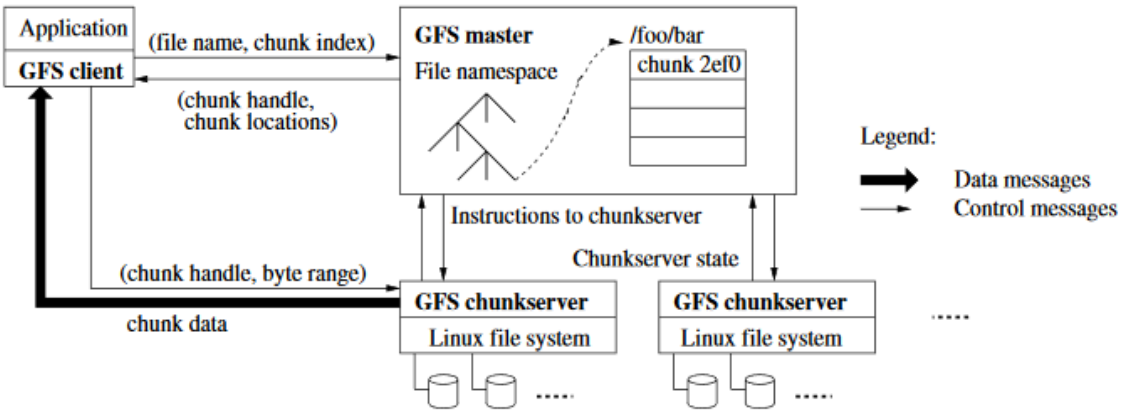

Abbildung 5: Read-Operation im GFS[14]

3 Security, Safety, Availability

3.1 IoT sichere Kommunikation

Die sichere Kommunikation zwischen IoT Sensoren und einer Verwaltungsplattform ist wichtig, um das Verfälschen oder Abfangen von übertragenen Daten zu erschweren. Dies ist vor allem ein Problem bei kritischen Daten, wie den Sensoren in einem Bojen Netzwerk. Durch Verfälschung der Daten könnte eine Tsunami-Warnung ausgelöst werden, welche eine Evakuierung notwendig machen würden.

Durch das signieren und verschlüsseln der Meldungen von IoT Netzwerken kann eine Verfälschung von Daten erheblich erschwert werden. Da IoT Geräte meist nicht über viel Rechenleistung verfügen, sollte hier eine Methode zur Verschlüsselung verwendet werden welche wenig Rechenleistung benötigt. Für diesen Anwendungsfall empfiehlt sich daher SSL oder IPSec. Eine andere Möglichkeit wäre es eine UDP Verbindung zu verwenden und diese mit DTLS zu sichern. Zur Übertragung der Nachrichten wird REST verwendet. [15][16]

SSL/TLS: wird verwendet um HTTP Anfragen abzusichern. Falls SSL/TLS Verwendet wird, dann wird dies HTTPS genannt. Das Protokoll wurde unter dem Namen SSL bis zu Version 3.0 entwickelt und danach in TLS Umbenannt. Bei der Umbenennung wurde die Versionsnummer auf 1.0 gesetzt. TLS steht für Transportation Layer Security. Zum Zeitpunkt des Verfassens ist TLS 1.2 aktuell. TLS verwendet einen asymmetrischen Schlüsselaustausch und für die Datenübertragung eine symmetrische Verschlüsselung. Wichtig bei TLS ist die verwendete Cypher Suite. Diese bestimmt die Methode des Schlüssel Austausches und die der symmetrischen Verschlüsselung. Eine Cypher Suite besteht aus einem Key Exchange Algorithm, einem Authentication Algorithm, einem Symmetric Encryption Algorithm, der Keylänge des Symmetrischen Schlüssels in Bit und einem Hash Algorithmus. Ein Beispiel einer solchen Suite wäre TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256. Diese Verschlüsselung wird aktuell von Google verwendet. Ein großes Problem der Cypher Suites ist die Kompatibilität mit den Geräten. Nicht jedes Gerät unterstützt die neueste Verschlüsselungsmethode. Aus diesem Grund muss hier ein Kompromiss zwischen Sicherheit und Usability gefunden werden. Wenn die beste Verschlüsselung verwendet wird, aber nur noch 10% der Clients zugreifen können ist das schlecht. Wenn jedoch viele Cypers akzeptiert werden steigt das Sicherheitsrisiko und das ist ebenfalls schlecht. Falls ein geschlossenes

Netzwerk von IoT Geräten verwendet wird kann die sicherste Verschlüsselung verwendet werden, da hier auf jedem Gerät die Cypher Suite dafür auch installiert werden kann. Im Falle eines offenen Netzwerkes ist dies aufgrund von genannten Kompatibilitätsproblemen nicht möglich. [17][18][19]

Lightweight IPSec: Eine andere Möglichkeit wäre es lightweight IPSec zu verwenden. IPSec ist eine Form von VPN und sichert somit die Kommunikation zwischen Geräten. Es ist eine sichere Version des IP Protokolls und arbeitet auf Layer 3. Im Gegensatz zu IPSec Arbeitet TLS auf Layer 4 und sichert somit die Kommunikation zwischen Anwendungen bzw. Diensten. IPSec kann im Tunnelmode oder im Transportmode arbeiten. Im Tunnelmode wird das Datenpaket komplett gekapselt und in einem „Transportpaket" verpackt und verschlüsselt. Dieses Modell wird vor allem für Remote Access Verbindungen verwendet(Client zu Server). Im Transportmode werden Daten verschlüsselt, jedoch nicht gekapselt. Dadurch hat das Paket weniger Overhead. Dieser Mode wird vor allem verwendet um z.B. zwei Firmenstandorte zu verbinden. Um die Verbindung lightweight zu machen kann 6LoWPAN-IPSec verwendet werden. Da die Daten hier mit 802.15.4 übertragen werden, beträgt die maximale Paketgröße 127 Byte. 6LoWPAN komprimiert die Header der Übertragung um mehr Platz für die Daten im Paket zu bieten. IPSec sollte im Transportmode verwendet werden, da dieser wie bereits erwähnt weniger Overhead hat. [20][21][22]

3.2 Firewall Konzepte

Es gibt zwei wichtige Kategorien von Firewalls: Paketbasierte und Applikationsbasierte Firewalls. [23][24]

Paketbasierte Firewalls arbeiten auf einer niedrigeren Netzwerkebene als Applikationsbasierte. Sie prüfen ob Computer A mit Server A kommunizieren darf. Dazu werden einige Daten des eingehenden Paketes analysiert. Zu diesen gehören z.B. die Source-IP, Destination-IP, Port und Protokoll. Durch diese Daten kann die Firewall anhand von vom Administrator festgelegten Regeln prüfen, ob das Paket zum Server weitergeleitet werden darf oder nicht. Falls das Paket nicht erlaubt ist kann es entweder ein DENY oder ein DROP bekommen. Im Falle von DENY bekommt der Absender eine Fehlermeldung. Bei DROP wird das Paket einfach weggeschissen. Es wird zwischen stateless und stateful Firewalls unterschieden. Bei stateless Firewalls wird jedes einzelne Paket nach Regelkonformität geprüft. Statefull Firewalls auch Smart Firewalls genannt erkennen Paketstreams und können auch Muster in den Streams

erkennen. Die States sind dieselben wie beim TCP Handshake. Einfache und detaillierte Logs sind wichtig, um neue Muster erkennen und neue Regeln hinzufügen zu können. [23][24]

Im Gegensatz zu Paketfilter-Firewalls sehen sich Application Firewalls(Layer 7) die Pakete genauer an. Diese Analysieren unter anderem die Layer 7 Daten des Paketes. Sie können somit Befehle welche einer Applikation auf einem Server geschickt werden vorher überprüfen und diese auch ablehnen. Sie können auch zur Absicherung einer Authentifizierung verwendet werden, indem sie nur bestimmten Usern erlauben einen Befehl zu verwenden. Sollte ein nicht autorisierter User einen solchen Befehl an den Server schicken, so lässt die Firewall das Paket nicht einmal zum Server. Dadurch, dass alle Pakete genauer inspiziert werden können auch viel genauere Logs erstellt werden. Somit lassen sich Angriffsmuster leichter erkennen. Da die Firewalls die Pakete genauer analysieren, sind sie auch um einiges langsamer. [25]

Firewalls können entweder einfache Applikationen sein, welche auf Endgeräten oder Servern installiert werden, oder sie sind dedizierte Hardware Produkte. Ein Beispiel für eine Software Firewall wäre iptables. Diese Software ist das Standardpaket für Firewalls auf Linux Systemen. Eine Hardware Firewall schützt ein gesamtes Netzwerk. Sie wird zwischen den Servern bzw. Endgeräten und dem Uplink positioniert. Diese Produkte sind meistens dazu gedacht in einem Rack montiert zu werden und kosten ab 500€ aufwärts. Der Vorteil einer Hardware Firewall ist es ein ganzes Netzwerk zu schützen. Dadurch ist der Konfigurationsaufwand für einen Administrator geringer. Es sollte jedoch darauf geachtet werden diese Firewalls redundant auszuführen, da ansonsten bei einem Ausfall alle Clients/Server keinen Internetzugriff mehr haben. Hardware Firewalls werden meist nur in Unternehmen eingesetzt. Sie können dadurch, dass sie dedizierte Geräte zum Analysieren und Filtern von Traffic sind komplexere Angriffe einfacher abwehren. [26]

3.3 DDoS

DDoS oder Distributed Denial of Service Attack ist eine Form eines Netzwerk Angriffes. Bei einem DDoS Angriff wird versucht, den Server durch eine Vielzahl von Anfragen zu überlasten. Wenn ein Server viele tausende von Anfragen bekommt, dann kann er die Anfragen der echten User nicht mehr beantworten, da er ja auch auf alle Anfragen der falschen Angriffsuser antworten muss. Oft werden für die Angriffe Botnets verwendet. In den letzten Jahren sind die Angriffe immer größer und stärker geworden. Dies hat

vor allem mit den Amplification Attacks und dem IT Schwarzmarkt zu tun. Seit einiger Zeit werden Botnets auch vermietet. Man kann bereits um 150$ einen einwöchigen schwachen DDoS Angriff kaufen. Es gibt täglich mehr als 2000 DDoS Angriffe und diese sind ca. für 1/3 aller Website-Ausfälle verantwortlich. Es gibt vier Klassen von „einfachen" DDoS Angriffen[27]:

TCP Connection Attack: Bei dieser Attacke wird einfach versucht Loadbalancer und Firewalls zu überlasten, indem immer neue TCP Anfragen geschickt werden. Es wird also versucht die Server mit dem TCP Verbindungsaufbau und Erhalt zu überlasten. [27]

Volumetric Attacks: Hier wird versucht innerhalb des Servernetzwerkes oder im Uplink Netzwerk der Server viel Traffic zu verursachen. Somit wird der Uplink einfach verstopft und die Server antworten langsam oder garnicht. [27]

Fragmentation Attack: Bei diesem Angriff werden TCP oder UDP Paketfragmente an den Server geschickt. Dieser versucht die Pakete wieder zu reparieren bzw. den Stream wiederherzustellen. Dadurch wird die Performance des Servers beeinträchtigt. [27]

Application Attacks: Diese Angriffe attackieren die zur Verfügung gestellte Applikation. Dazu wird meist eine rechenaufwendige Funktion der Applikation missbraucht. Sie brauchen nur wenig Traffic und sind daher schwer zu finden. [27]

Bei einer DDoS Amplification Atttack werden schwachstellen in Protokollen ausgenutzt, um den Angriff zu verstärken. In diesen Angriffen wird meist UDP verwendet, da dieses keinen Handshake erfordert. Da kein Handshake besteht, kann die Source-IP der Anfrage gespooft werden. Durch solche Amplification Attacks kann der Angreifer mit einem kleinen Botnet einen größeren Schaden anrichten. Aktuell sind die NTP und DNS Reflection Amplification Attacks beliebt. [28][29]

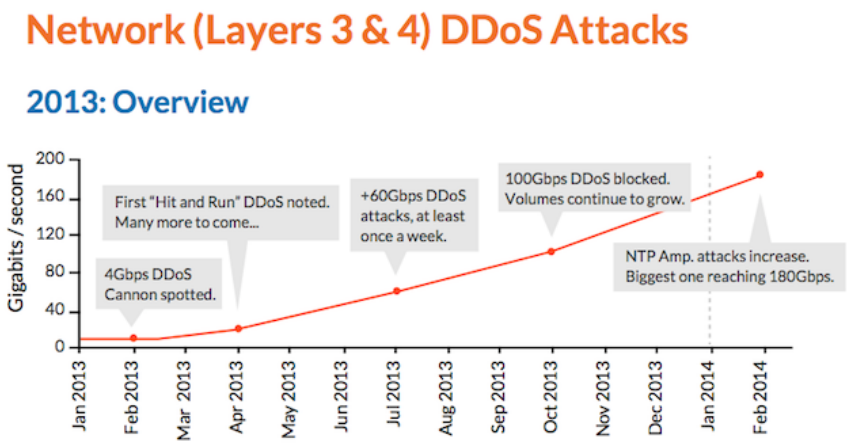

Abbildung 6: DDoS Trend 2013 [30]

DNS Amplification: Bei DNS Amplification Attacks werden Open Resolvers ausgenutzt. Das sind öffentliche rekursive DNS Server. Rekursive DNS Server lösen eine Domain auf eine IP auf. Einige Domains haben jedoch komplexe und lange Domain Einstellungen welche gut für Angriffe verwendet werden können. Bei diesem Angriff sende ich DNS Anfragen an einen Open Resolver für eine möglichst komplizierte Domain. Als Absenderadresse trage ich aber die Adresse meines Opfers ein. Somit antwortet der DNS Server an mein Opfer. Ich selbst habe nur eine kleine DNS Anfrage versendet, jedoch war die Antwort viel länger. Es kann hier die 50-60 fache Antwortgröße durch diesen Angriff erzielt werden. Zum Beispiel kann eine 64 Byte Anfrage an isc.org geschickt werden. Die Antwort ist jedoch 3233 Bytes lang.[28]

NTP Amplification: NTP ist das Internet Zeit Protokoll. Es wird verwendet um die eigene Uhr mit der eines Zeitservers abzugleichen. Ältere Versionen des Protokolls unterstützen einen bestimmten Befehl, welcher eigentlich zum Finden von Fehlern durch Administratoren gedacht war. Dieser Befehl heißt „monlist" und gibt die Verbindungsinformationen der letzten 600 Clients zurück. Der Angreifer schickt also eine Anfrage an einen NTP Server mit veralteter Software und gibt als Source-IP die des Opfers an. Der Server gibt nun die Letzen 600 verbundenen Clients zurück und schickt diese an das Opfer. Dadurch lassen sich Angriffe um einen hohen Faktor skalieren, je nachdem wie viele Clients in der Liste waren. [29]

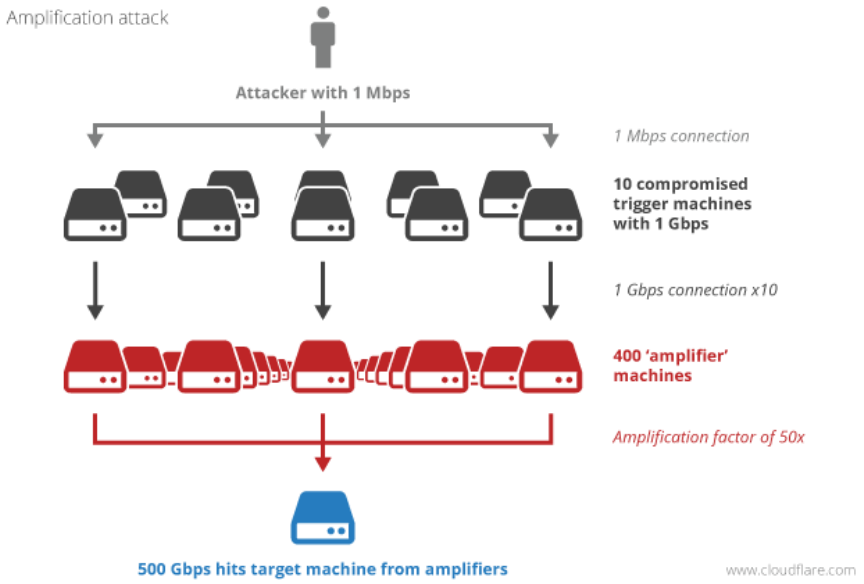

Abbildung 7: DDoS Amplification [31]

Es ist äußerst schwierig geworden sich gegen DDoS Angriffe abzusichern. Die immer stärker werdenden Angriffe können so gut wie jedes Netzwerk lahmlegen. Einen gewissen Schutz bieten Hardware Firewalls. Diese können Angriffe bis zu einigen Gbit Größe abwehren. Die Anschaffungskosten für solche Geräte sind allerdings sehr hoch. Immer mehr Hosting Provider inkludieren Netzwerkschutz durch Hardware Firewalls in ihren Hosting Paketen, allerdings meist nur bis 5Gbit. Falls der Angriff jedoch zu stark wird, nehmen einige Hoster den Server vom Netz oder routen den Traffic direkt zum Server durch ohne einen Versuch ihn abzuwehren. Eine andere Möglichkeit sich gegen DDoS Angriffe zu schützen ist es ein CDN zu verwenden. Diese verwenden Anycast um die Anfragen an ihr nächstes Rechenzentrum weiterzuleiten. Dadurch wird der Angriff auf viele Rechenzentren aufgeteilt und müsste außerordentlich groß sein um Probleme zu verursachen. [31]

3.4 VPN

Ein VPN wird wie bereits in Punkt 3.1 erwähnt dazu verwendet, sich von Zuhause aus zum Firmennetzwerk zu verbinden oder um Firmenstandorte zu verbinden. Es kann auch dazu verwendet werden um die eigene Verbindung in z.B. einem öffentlichen WLAN abzusichern. Eine wichtige Rolle bei VPNs spielt die Sicherheit bzw. Verschlüsselung der Übertragung. Es gibt mehrere VPN Protokolle, jedoch sind nicht mehr alle sicher. Am sichersten ist OpenVPN, welches TLS verwendet um Daten zu verschlüsseln. L2TP oder IPSec bieten auch noch gute Sicherheit, PPTP sollte jedoch nicht mehr verwendet werden. [32]

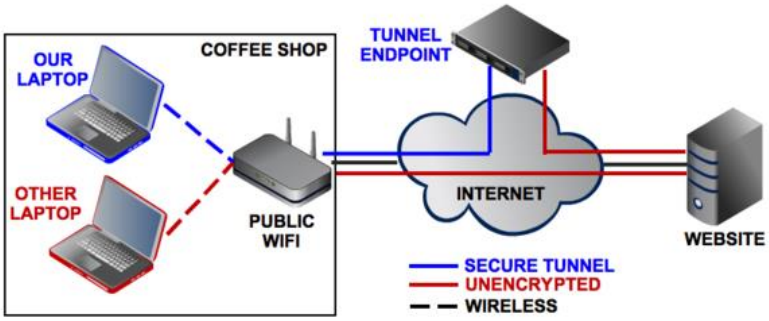

Abbildung 8: VPN Verwendung im öffentlichen WLAN [32]

Um eine VPN Verbindung möglichst sicher zu gestalten sollte immer die derzeit sicherste Anmeldemethode verwendet werden. Am besten wäre es auch, nicht nur Passwörter zur Anmeldung zu verwenden. Eine kombinierte Authentifizierung z.B. mit einer SmartCard würde die Anmeldung besser absichern. Es sollte auch immer die beste von den Endgeräten unterstützte Verschlüsselungsmethode verwendet werden. [33]

4 Authentication, Authorization, Accounting

4.1 Benutzerverwaltung mit LDAP

Lightweight Directory Access Protocol ist ein Verzeichnisdienst welcher den vorherigen Standard X500 ersetzt. LDAP selbst ist eigentlich nicht wirklich lightweight, außer man Vergleicht es mit X500 welches wesentlich komplexer war. LDAP kann als objektorientierte Datenbank angesehen werden, in welcher die Objekte in einer Baum-Struktur abgespeichert werden. Jedes Objekt kann n Attribute haben und von mehreren Klassen erben. Es gibt drei Arten von Klassen [34]:

Structual Class: Ein Objekt muss eine und kann maximal eine Structural Class haben. Diese mappt das Objekt auf einen realen Gegenstand oder eine Person. Die Klasse kann nicht geändert werden, außer das Objekt wird gelöscht und neu erstellt. [34]

Auxiliary Class: Ein Objekt kann Attribute von Auxiliary Classes erben. Es kann von einer oder mehrerer Klassen erben. Diese können beliebig hinzugefügt oder entfernt werden. [34]

Abstract Class: Ist eine Vorlage für andere Klassen und kann nicht direkt instanziert werden. [34]

Diese Klassen werden einem Objekt durch das `objectClass` Attribut zugewiesen. LDAP selbst hat einige einfache Klassen, es können aber auch neue erzeugt werden. Ein Objekt hat immer eine genaue Adresse welches es eindeutig identifiziert(Distinguished Name, `dn`). Dieser Name wird aus dem Baum bis zum Objekt und einem eindeutigen Key gebildet. Bei Personen könnte z.B. der `cn` also der volle Name verwendet werden. Eine Adresse zu so einem Personenobjekt wäre dann `cn=David Pashley,ou=People,dc=example,dc=com`. Die Bezeichnung `dc`, Domain Component beschreibt die Domain zu welcher ein Objekt gehört. Die Organizational Unit also `ou` beschreibt eine Gruppe von Objekten zu welcher dieses Objekt gehört. Um nun einen User für ein POSIX System in LDAP Abzubilden sollte ein Objekt mit der `objectClass` person(Structural), inetOrgPerson(Auxiliary) und posixAccount(Auxiliary) erstellt werden. Mit dieser Klasse kann eine Person und ein POSIX Account abgebildet werden. Die posixAccount Klasse fügt Attribute wie uid, homeDirectory und loginShell hinzu. [34]

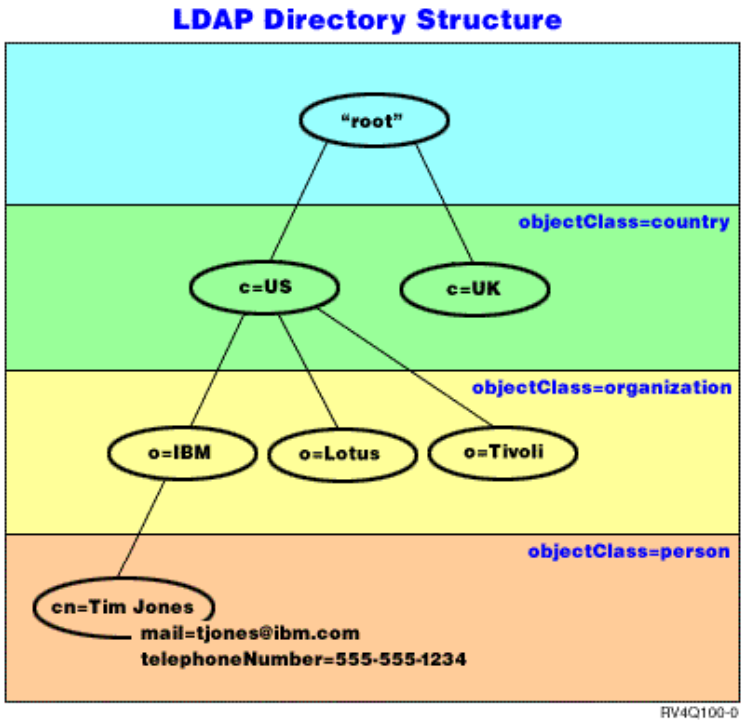

Abbildung 9: LDAP Beispiel [35]

4.2 Sicherstellen der User-Identität

In der digitalen Welt werden jede Menge von Anmeldeinformationen gebraucht. Meist werden nur Passwörter verwendet, es gibt jedoch auch andere unterstützende Verfahren. Der große Nachteil von Passwörtern ist, dass viele Menschen für alle Accounts das gleiche Passwort verwenden. Dieses ist dann oft auch noch ein einfaches Passwort. Das Problem an solch einfachen Passwörtern ist, dass sie sehr einfach geknackt werden können. Wenn eine Person für jeden Account dasselbe Passwort verwendet, dann muss nur die Datenbank eines Serviceanbieters geknackt werden und alle Accounts dieser Person sind unsicher. [36]

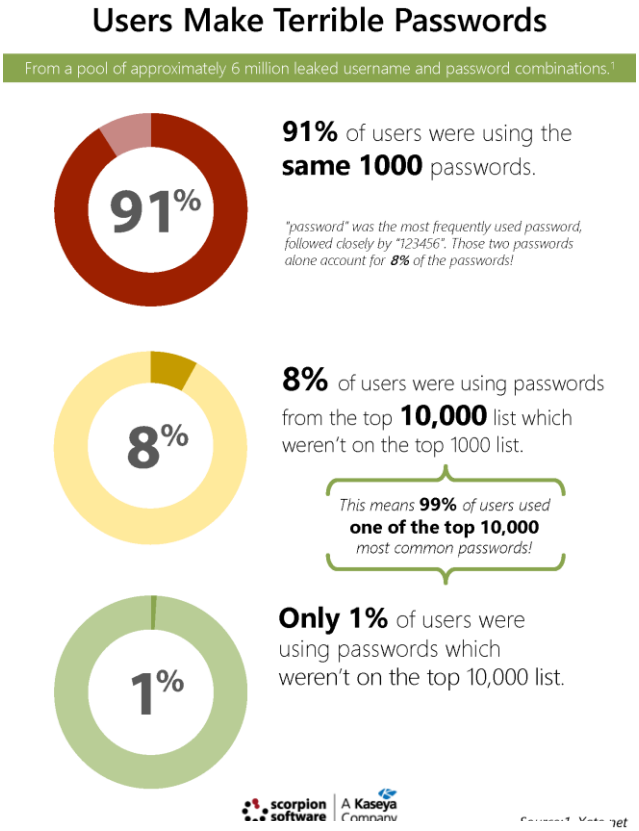

Abbildung 10: Passwort Statistik [36]

Aus diesem Grund wird immer öfter 2-Factor Authentifizierung verwendet. Das bedeutet, dass mehr als nur das Passwort benötigt wird um sich Anzumelden. Es wird also eine Kombination aus Wissen und Besitz erfordert um sich anzumelden. Auf Websites wird häufig das One Time Password(OTP) Verfahren verwendet. Bei diesem Verfahren wird nach der Eingabe des Passwortes eine SMS an den User geschickt in welcher ein Code steht. Dieser Code wird jedes Mal neu generiert wenn ein Anmeldeversuch stattfindet. Um sich nun Anzumelden werden also das Passwort und das Handy des Accounteigentümers benötigt. Es kann auch keine SMS sondern eine App verwendet werden. Diese generiert meist alle 30 Sekunden ein neues OTP. [37]

In Unternehmen in welchen hohe Sicherheitsstandards notwendig sind wie z.B. in Banken werden oft SmartCards verwendet. Zur Anmeldung an einem Computer oder Software werden dann eine Smartcard und ein PIN benötigt. Bankomatkarten sind auch SmartCards. [38]

4.3 RDP

Das Remote Desktop Protocol kann verwendet werden um auf entfernte Geräte zuzugreifen. Hierbei wird der gesamte Bildschirm des entfernten Gerätes übertragen und es kann so gearbeitet werden als säße man direkt davor. Das Protokoll wurde von Microsoft entwickelt. Eine Alternative zu RDP ist VNC. [39]

5 Desaster Recovery

5.1 Client Rollback

In größeren Unternehmen mit mehreren hundert PCs wird es schwierig jeden einzeln zu konfigurieren und zu warten. Aus diesem Grund werden oft zentrale Images verwendet, welche dann auf jeden Client gespielt werden. Somit reduziert sich der Aufwand enorm. Es wird ein zentrales Image aktuell gehalten und benötigte Software wird installiert. Dieses Image wird dann auf jeden Client gespielt. Somit hat jeder Client denselben Softwarestand und es ist um einiges einfacher Probleme zu beheben. Falls nun ein größeres Problem auf einem Client auftritt kann einfach das zentrale Image neu eingespielt werden. Die Daten des Users werden meist auf Netzwerklaufwerken abgelegt, um Datenverlust zu verhindern. [40]

5.2 Standby

Beim Betreiben eines digitalen Services ist es wichtig, mögliche Ausfallszenarien zu berücksichtigen, da falls der betriebene Service nicht zuverlässig ist, Kunden nicht bleiben werden. Um bei Fehlern schnell den Service wieder bereitzustellen, sollten die Standby Modelle beachtet werden. Bei den Standby Modellen wird ein zweites System, welches das erste ersetzen könnte bereitgehalten. Hierbei gibt es drei Standby Varianten. [41]

Hot Standby: Hierbei wird die Hardware und Software komplett auf ein zweites System gespiegelt. Beide Systeme synchronisieren ständig ihren Datenbestand. Fällt nun das Primärsystem aus so kann fast sofort das zweite System den Dienst übernehmen. Beide Systeme sind immer aktiv. [41]

Warm Standby: Hierbei wird das Sekundärsystem nur in periodischen Abständen mit dem Primärsystem synchronisiert. Dadurch entsteht hier weniger Overhead. Beide Systeme sind immer aktiv. [41]

Cold Standby: Hier werden die Systeme nicht aktiv synchronisiert. Dies muss händisch geschehen. Dadurch dauert es etwas länger bis das System wieder richtig arbeitet, es ist in der Erhaltung jedoch billiger. [41]

5.3 Desaster Recovery Plan

Hier gibt es ein Standard Modell von IBM namens SHARE [42]:

Tier 0: No off-site data – Possibly no recovery

„Unternehmen mit einer Tier 0-Lösung haben keinen Disaster Recovery Plan. Sie haben keine gespeicherten Informationen über das System, keine Dokumentation und auch keine Backups. Die Zeit, die benötigt wird um ein solches System wiederherzustellen ist unvorhersehbar, es kann sogar sein, dass es unmöglich ist zum Normalzustand zurückzukehren.“ [42]

Tier 1: Data backup with no hot site

„Tier 1-Systeme erhalten regelmäßig ein Backup, und lagern diese an einem sicheren Ort, der sich außerhalb des eigenen Hauses (der eigenen Infrastruktur) befindet. Dies ist notwendig, da vor einem Desaster auch Backups mit lokalem, nicht redundanten Speicherort nicht sicher sind. Diese Methode Backups zu transportieren heißt PTAM, ausgeschrieben ’Pick-up Truck Access Method’. Sprich, das Backup ist so schnell greifbar, wie der Transport vom Aufbewahrungsort dauert. Abhängig davon, wie oft Sicherungen gemacht und versendet werden müssen Unternehmen einige Tage bzw. Wochen Datenverlust inkaufnehmen, dafür sind die Sicherungsdateien geschützt außerhalb des Geländes aufbewahrt.“ [42]

Tier 2: Data backup with a hot site

„Bei Verwendung von Tier 2 werden ebenso regelmäßige Sicherungen vorgenommen, auf langlebigen Speichermedien wie etwa Tapes. Das wird kombiniert mit eigener Infrastruktur außerhalb des eigenen Geländes (genannt ’Hot Side’), von welchen die Backups rückgelesen werden im Falle eines Desasters. Diese Lösung benötigt immer noch einige Stunden oder Tage zur Wiederherstellung, jedoch ist die Gesamtdauer besser vorhersehbar.“ [42]

Tier 3: Electronic vaulting

„Der Ansatz Tier 3 baut auf Tier 2 auf. Hinzufügend dazu werden kritische Daten elektronisch abgekapselt von den weniger prioren. Die Abgekapselten sind üblicherweise aktueller als die Daten, die via PTAM abgelegt werden. Als Ergebnis ist hier weniger Dateiverlust, sollte eine Katastrophe eintreten. Einrichtungen, die 'Electronic Remote Vaulting' bereitstellen, verfügen über Hochgeschwindigkeitsanbindungen und entweder physische oder virtuelle Sicherheitstape-Lesegeräte, mit automatisierter Sicherungsbibliothek beim entfernten Standort. Als praktischen Beispiel zur Implementierung dienen IBMs Peer-to-Peer TotalStorage Virtual Tape Server oder Oracles VMS Clustering." [42]

Tier 4: Point-in-time copies

„Tier 4-Lösungen werden oft von Unternehmen verwendet, die hohen Wert auf Datenkorrektheit und schneller Wiederherstellung legen als die unteren Stufen bereitstellen. Eher als das Auslagern von Speichertapes wie bei 0-3 gegeben, integriert diese Stufe Sicherungen auf Basis von Disks (also Festplatten). Immer noch sind mehrere Stunden Datenverlust möglich, jedoch ist es einfacher Point-in-Time-Backups (PiT, jeweils zu einem festgelegten Zeitpunkt) zu erstellen mit einer höheren Frequenz als Tape-Sicherungen, sogar wenn elektronisch gekapselt." [42]

Tier 5: Transaction integrity

„Tier 5 wird verwendet, wenn es zwingend erforderlich ist, dass Daten konsistent sind zwischen Produktivsystem und dem Wiederherstellungs-Remoteserver. Bei einem solchen Aufbau kommt es zu kaum bis gar keinem Datenverlust, aber die Verfügbarkeit dieser Funktionalität ist stark abhängig von der verwendeten Implementierung." [42]

Tier 6: Zero or near-zero data loss

„Tier 6 hält das höchste Maß an Datenrichtigkeit aufrecht. Dieser Ansatz wird eingesetzt von Systemen, in denen wenig oder gar kein Datenverlust vertretbar ist, und wo Daten für den weiteren Gebrauch schnell wiederhergesellt werden müssen. Solche Lösungen haben keine Abhängigkeit von Anwendungen oder Mitarbeitern, um Datenkonsistenz zu gewährleisten. Tier 6 erfordert eine Form von Disk Mirroring zu einem Remoteserver, hierfür gibt es verschiedenste synchrone oder asynchrone Implementierungen von vielen Herstellern. Jede Herangehenweise ist leicht anders, mit verschiedenen Möglichkeiten und anderen Recovery Point/Recovery Time-Anforderungen. In den manchen Fällen wird jedoch auch eine Art von Tape-Speicherung benötigt, abhängig von dem erforderlichen Speicherplatz, der von Backup in Anspruch genommen wird.“ [42]

Tier 7: Highly automated, business integrated solution

„Das höchste Level nimmt all die Hauptkomponenten von Tier 6 zusammen und fügt die Intgration von Automation hinzu. Das erlaubt Tier 7 damit auch die Datenkonsistenz, die bei Tier 6 gegeben ist. Desaster werden automatisch erkannt von Geräten außerhalb des eigenen Computersystems. Außerdem wird die Wiederherstellung automatisch ausgeführt, was sozusagen den kompletten Wiederherstellungsprozess bei System und Anwendungen beschleunigt, und diese schneller und zuverlässiger laufen lässt als es überhaupt möglich wäre durch händische Prozeduren. Die Ausfälle belaufen sich auf wenige Minuten oder Sekunden.“ [42]

6 Algorithmen und Protokolle

6.1 SNMP

Das Simple Network Management Protokoll dient dazu, Geräte über ein Netzwerk zu überwachen und zu konfigurieren. Server und Netzwerkhardware haben meist extra Management Ports für eine SNMP Anbindung. Die Administration der Geräte wird dadurch erheblich einfacher, da auch mehrere Geräte gleichzeitig angesprochen werden können. Das Protokoll ist sehr einfach gehalten, um auch von Sensoren wie Feuermeldern verwendet werden zu können. Sicherheitstechnisch ist das Protokoll eher schlecht. Um SNMP besser verstehen zu können, müssen die einzelnen Komponenten genauer erläutert werden. [43]

Agent: Ist das Netzwerkgerät welches mit SNMP gesteuert werden soll. [43]

Client: Vom Client aus werden per SNMP einer oder mehrere Agents gesteuert. [43]

OID: Der Object Identifier identifiziert jedes SNMP fähige Gerät. Diese werden bis zu einer bestimmten Tiefe von der IANA vergeben. [43]

MIB: Die Management Information Base enthält die Zugriffsmöglichkeiten für ein Gerät. Diese wird vom Client benötigt, um zu wissen welche Daten abgefragt und welche Befehle gegeben werden können. [43]

Community: Community ist ein String, welcher quasi die Berechtigungsstufe des Client beschränkt. [43]

Der SNMP Befehlssatz ist sehr einfach. Im Prinzip gibt es nur verschiedene Versionen von Getter & Setter Methoden. Es kann auch eine sogenannte Trap erstellt werden. Das bedeutet, dass ein Gerät selbstständig eine Nachricht versendet, sobald ein definiertes Event passiert. [43]

Bisher gibt es von SNMP Version 1 und Version 2. Version 2 bietet mehr Befehle um Daten abzurufen und hat eine einfache Authentifizierung, welche jedoch leicht ausgetrickst werden kann. SNMPv3 sollte sicherer sein, der RFC2273 des Protokolls ist jedoch nur als PROPOSED STANDARD definiert. Das bedeutet, dass SNMPv2c nach wie vor als Standard verwendet wird. [43]

6.2 Container & Containerverwaltung

In den letzten Jahren wurden Container immer populärer. Dies hatte vor allem mit dem Release von Docker zu tun. Doch wieso sind Container so beliebt? Container sind verglichen mit normaler Virtualisierung effizienter. Normalerweise werden auf einem Server mehrere VMs betrieben. Der Server hat ein Betriebssystem installiert welches Virtualisierung unterstützt. Die einzelnen VMs haben jeweils auch ein Betriebssystem installiert. Und hier liegt das Problem. Durch die doppelte Schicht von Betriebssystemen entsteht ein großer Overhead. Container teilen sich ein Betriebssystem, sind aber trotzdem isoliert voneinander. Dadurch benötigen sie weniger Ressourcen und es können mehr Container auf einem Server betrieben werden. [44]

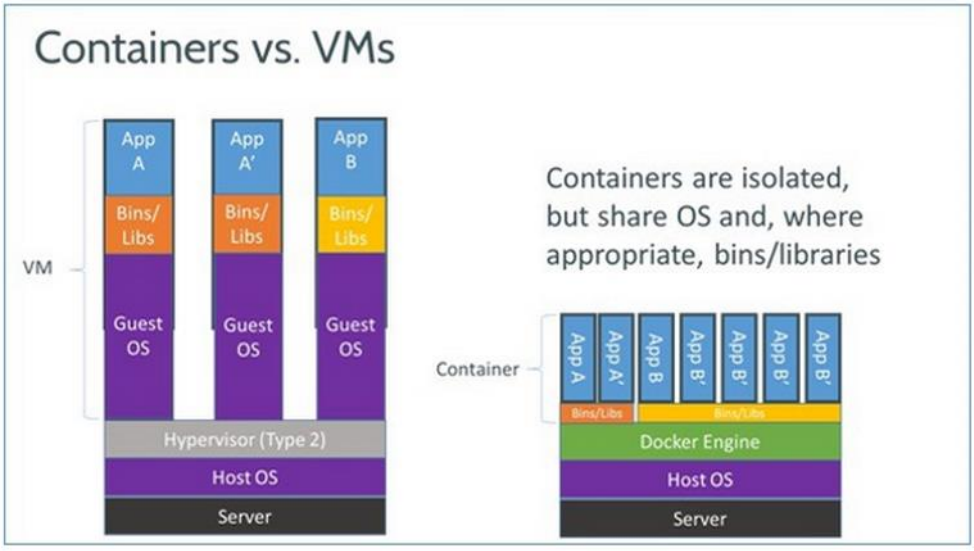

Abbildung 11: VM vs Container [44]

Dadurch, dass Container kein eigenes OS mehr haben, sind sie auch einfacher zu verwalten. Sie können viel schneller und einfacher erstellt, geändert oder gelöscht werden. Die Container brauchen auch weniger Speicherplatz. Container können auch besser skaliert werden. Für einen Anwendungsfall z.B. ein Webserver wird ein Container erstellt und konfiguriert. Das Image dieses Containers wird dann in eine Imagedatenbank geladen. Wenn nun der Webserver ausgelastet ist kann einfach ein neuer Container erzeugt werden. Noch einfacher wird das Ganze mit einem Management Tool wie Kubernetes. Hier ist es möglich, ab einem bestimmten Auslastungsgrades eines Servers einen neuen Container erstellen zu lassen. Container, welche nicht mehr richtig funktionieren werden automatisch gelöscht und neu provisioniert. [45]

6.3 Heartbeat

Ein Heartbeat ist ein Signal das die Funktionalität von Geräten überwacht. Sie werden hauptsächlich bei Clustern oder Loadbalancern eingesetzt. Sie überwachen z.B. ob der Webserver noch aktiv ist. Falls er das nicht ist, bekommt er vom Loadbalancer keine weiteren Clients mehr zugewiesen. In Clustern bzw. Failover Konfigurationen überprüft ein Management Sever die Funktionalität der angeschlossenen Server. Falls der Hauptserver ein Problem hat schaltet der Management Server auf den Slave Server um. [46]

7 Konsistenz und Datenhaltung

7.1 Samba & NFS

In Firmen werden wie bereits erwähnt oft zentral erstellte Images verwendet. Um die Daten nicht zu gefährden, werden diese oft auf Netzwerklaufwerken gespeichert. Es gibt hier zwei unterschiedliche Möglichkeiten. NFS kann verwendet werden, wenn das Netzwerk eher klein ist, da bis zu Version 4 fast keine Sicherheitsfeatures integriert waren. Seit Version 4 gibt es die Möglichkeit Kerbereos zu integrieren. NFS ist Betriebssystem unabhängig. [47]

Samba welches seit kurzem CIFS heißt bietet mehr Features zur Verwaltung, da es einfach an ein Active Directory angebunden werden kann. Unter Windows wird CIFS automatisch unterstütz, unter Linux muss der Samba Dienst installiert werden. [48]

7.2 Storage Cluster

Die heute verfügbaren Cloud Anbieter müssen mehrere Petabyte an Daten speichern. Doch wie ist es möglich, Speicherplatz so billig anzubieten? Ein Teil des Geheimnisses liegt in Algorithmen, welche doppelte Daten erkennen. Diese verwenden Hashfunktionen um Datenduplikate zu erkennen. Wird nun ein Duplikat versucht zu speichern, dann speichert der Algorithmus anstatt der Daten nur eine Adresse zur eigentlichen Datei. Somit können Unmengen von Speicherplatz gespart werden. Komplexere Algorithmen teilen die Daten nicht in Dateien sondern in Speicherblöcken. Dadurch kann noch mehr Platz gespart werden. Dieses Verfahren wird Data Deduplication genannt[49]

Abbildung 12: Data Deduplication [49]

Doch wie werden die übrig gebliebenen Daten abgespeichert. Der Trend in der Server Branche geht weg von Storage direkt im dazugehörigen Server Blade. Stattdessen werden immer mehr Storage Blades verwendet. Das sind Blades welche meist 4 RU oder mehr hoch sind und eine Vielzahl von Festplatten/SSDs betreiben. Der Storage für die Server wird dann übers Netzwerk zugeteilt. Dieses Verfahren ist Platzsparender und einfacher zu warten. Wirklich große IT Unternehmen wie Google verwenden jedoch schon seit langem keine Server Hardware mehr, da diese viel zu teuer ist. Google verwendet normale Consumer Hardware und berechnet auch mit ein, dass diese häufiger ausfällt. [50]

7.3 Dateisysteme für Storage Cluster

siehe Hadoop und GFS in Kapitel 1.5

8 Quellen

[1], Felix Kling, JSON and XML comparison, http://stackoverflow.com/a/4862511, zuletzt besucht: 25.05.16

[2], Yegor Bugayenko, Stop Comparing JSON and XML, http://www.yegor256.com/2015/11/16/json-vs-xml.html, zuletzt besucht: 25.05.16

[3], Carlos Sanchez, Scaling Docker with Kubernetes, https://www.infoq.com/articles/scaling-docker-with-kubernetes, zuletzt besucht: 25.05.16

[4], Motasem Aldiab, Public Cloud War: AWS vs Azure vs Google, http://cloudacademy.com/blog/public-cloud-war-aws-vs-azure-vs-google/, zuletzt besucht: 25.05.16

[5], Tsahi Levent-Levi, IOT Messaging – Should we Head for the Cloud or P2P?, https://bloggeek.me/messaging-iot-p2p/, zuletzt besucht: 25.05.16

[6], Revisionworld, Star Topology, http://revisionworld.com/sites/revisionworld.com/files/rw_files/star_topology_89.jpg, zuletzt besucht: 25.05.16

[7], Agnes Chau, Security Concerns of Peer-to-Peer Software, https://www.its.hku.hk/news/ccnews125/p2p.jpg, zuletzt besucht: 25.05.16

[8], Treelineinteractive, Hubs and Spokes, http://treelineinteractive.com/blog/wp-content/uploads/2015/09/Device-to-phone-to-cloud.jpg, zuletzt besucht: 25.05.16

[9], MIT, IP Addresses, Host Names, and Domain Names, https://ist.mit.edu/network/ip, zuletzt besucht: 26.05.16

[10], Ubuntuusers, SSH, https://wiki.ubuntuusers.de/SSH/, zuletzt besucht: 26.05.16

[11], TechTarget, DFS, http://searchwindowsserver.techtarget.com/definition/distributed-file-system-DFS, zuletzt besucht: 26.05.16

[12], Oracle, Project: OCFS2, https://oss.oracle.com/projects/ocfs2/, zuletzt besucht: 26.05.16

[13], Tutorialspoint, Hadoop - HDFS Overview, http://www.tutorialspoint.com/hadoop/hadoop_hdfs_overview.htm, zuletzt besucht: 26.05.16

[14], Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, The Google File System, https://static.googleusercontent.com/media/research.google.com/de//archive/gfs-sosp2003.pdf, zuletzt besucht: 26.05.16

[15], Stackoverflow Users, How to secure RESTful web services?, https://stackoverflow.com/questions/4817643/how-to-secure-restful-web-services, zuletzt besucht: 27.05.16

[16], Owasp Wiki, REST Security Cheat Sheet, https://www.owasp.org/index.php/REST_Security_Cheat_Sheet, zuletzt besucht: 27.05.16

[17], Marco Beierer, SSL, TLS und HTTPS erklärt, https://www.marcobeierer.at/wissen/ssl-tls-und-https-erklaert, zuletzt besucht: 27.05.16

[18], Stackexchange Users, Now that it is 2015, what SSL/TLS cipher suites should be used in a high security HTTPS environment?, https://security.stackexchange.com/questions/76993/now-that-it-is-2015-what-ssl-tls-cipher-suites-should-be-used-in-a-high-securit, zuletzt besucht: 27.05.16

[19], TheSprawl, TLS and SSL Cypher Suites, https://www.thesprawl.org/research/tls-and-ssl-cipher-suites/, zuletzt besucht: 27.05.16

[20], Administrator.de, IPSEC Protokoll - Einsatz, Aufbau, benötigte Ports und Begriffserläuterungen, https://www.administrator.de/wissen/ipsec-protokoll-einsatz-aufbau-ben%C3%B6tigte-ports-begriffserl%C3%A4uterungen-73117.html, zuletzt besucht: 27.05.16

[21], Christian Bockermann, Sichere Netzwerke mit IPSec, http://www.ping.de/~christian/IPSec.pdf, zuletzt besucht: 27.05.16

[22], Lancaster University, Securing Internet of Things with Lightweight IPSec, http://soda.swedish-ict.se/4052/2/reportRevised.pdf, zuletzt besucht: 27.05.16

[23], Dinesh Sharma, Firewall – Basic concepts, http://www.ebrahma.com/2015/04/firewall-basic-concepts/, zuletzt besucht: 27.05.16

[24], Ganesh Dutt Sharma, Packet Filtering Firewall: An Introduction, http://securityworld.worldiswelcome.com/packet-filtering-firewall-an-introduction, zuletzt besucht: 27.05.16

[25], Michael Cobb, Defending layer 7: A look inside application-layer firewalls, http://searchsecurity.techtarget.com/tip/Defending-layer-7-A-look-inside-application-layer-firewalls, zuletzt besucht: 28.05.16

[26], Michigan CyberSecurity, Hardware Firewall vs Software Firewall, https://www.michigan.gov/cybersecurity/0,4557,7-217--108698--,00.html, zuletzt besucht: 28.05.16

[27], Digital Attack Map, What is a DDoS Attack?, http://www.digitalattackmap.com/understanding-ddos/, zuletzt besucht: 28.05.16

[28], Matthew Prince, Deep Inside a DNS Amplification DDoS Attack, https://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack/, zuletzt besucht: 28.05.16

[29], Imperva Incapsula, NTP Amplification, https://www.incapsula.com/ddos/attack-glossary/ntp-amplification.html, zuletzt besucht: 28.05.16

[30], Brian Donohue, DDoS Threat Landscape Report, https://threatpost.com/ntp-amplification-syn-floods-drive-up-ddos-attack-volumes/105069/, zuletzt besucht: 28.05.16

[31], Cloudflare, DDoS Amplification, https://blog.cloudflare.com/content/images/illustration-amplification-attack-ph3.png, zuletzt besucht: 28.05.16

[32], Eric Geier, How (and why) to set up a VPN today, http://www.pcworld.com/article/2030763/how-and-why-to-set-up-a-vpn-today.html, zuletzt besucht: 28.05.16

[33], Martin Heller, 10 tips to secure client VPNs, http://www.computerworld.com/article/2547058/networking/10-tips-to-secure-client-vpns.html, zuletzt besucht: 28.05.16

[34], David Pashley, LDAP Basics, http://www.davidpashley.com/articles/ldap-basics/, zuletzt besucht: 29.05.16

[35], IBM, LDAP Basics, https://publib.boulder.ibm.com/html/as400/ic2924/info/RZAHYM04.HTM, zuletzt besucht: 29.05.16

[36], Tom Rizzo, Christmas Lists Are Consistent, Your Passwords Shouldn't Be..., http://insights.scorpionsoft.com/christmas-lists-are-consistent-your-passwords-shouldn%E2%80%99t-be, zuletzt besucht: 29.05.16

[37], SecurEnvoy, What is 2FA?, https://www.securenvoy.com/two-factor-authentication/what-is-2fa.shtm, zuletzt besucht: 29.05.16

[38], ITWissen, Smartcard, http://www.itwissen.info/definition/lexikon/Smartcard-smartcard.html, zuletzt besucht: 29.05.16

[39], Wikipedia, Remote Desktop Protocol, https://de.wikipedia.org/wiki/Remote_Desktop_Protocol, zuletzt besucht: 29.05.16

[40], Kevin Lo, Configure and Set Up Multiple Computers, http://www.techsoup.org/support/articles-and-how-tos/configure-and-setup-multiple-computers, zuletzt besucht: 29.05.16

[41], Sana Naveed Khawaja, Fault Tolerance in the Internet:Servers and Routers, http://www.sanog.org/resources/sanog8/sanog8-fault-tolerant-sana-tariq.pdf, zuletzt besucht: 29.05.16

[42], Erik Brändli & Michael Weinberger, Ausarbeitung Synchronisierung & Konsistenz

[43], Christian Eisner & Thomas Mayr, SNMP -Simple Network Monitoring Protokoll, http://www.dietmueller.at/download/05_SNMP.pdf, zuletzt besucht: 30.05.16

[44], Steven J. Vaughan-Nichols, What is Docker and why is it so darn popular?, http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/, zuletzt besucht: 30.05.16

[45], Google, What is Kubernetes?, http://kubernetes.io/docs/whatisk8s/, zuletzt besucht: 30.05.16

[46], ITWissen, Heartbeat, http://www.itwissen.info/definition/lexikon/heartbeat-Herzschlag.html, zuletzt besucht: 30.05.16

[47], Wikipedia, Network File System, https://de.wikipedia.org/wiki/Network_File_System, zuletzt besucht: 30.05.16

[48], Ubuntuusers, Samba, https://wiki.ubuntuusers.de/Samba/, zuletzt besucht: 30.05.16

[49], Chris Poelker, Data deduplication in the cloud explained, part one, http://www.computerworld.com/article/2474479/data-center/data-deduplication-in-the-cloud-explained--part-one.html, zuletzt besucht: 30.05.16

[50], Quora, How does Google store their data?, https://www.quora.com/How-does-Google-store-their-data, zuletzt besucht: 30.05.16

9 Abbildungsverzeichnis